

NO-A194 625

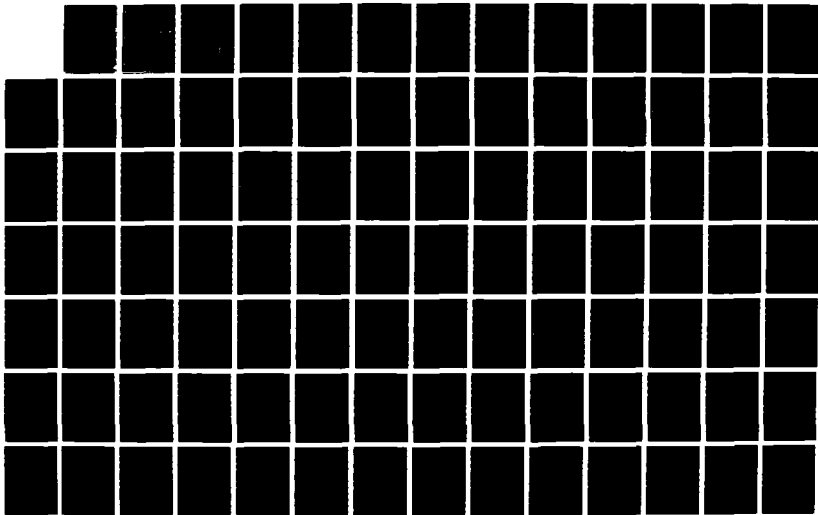
LRS-II: A SPECIALIZED KNOWLEDGE SYSTEM FOR LAUNCH
RESOURCE SCHEDULING(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
J E CRAWFORD DEC 87 AFIT/GE/ENG/87D-11

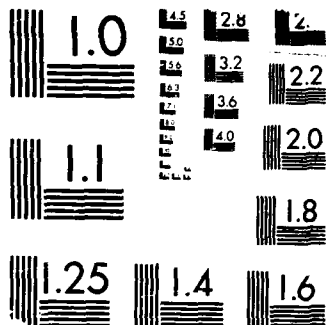
1/2

UNCLASSIFIED

F/G 5/1

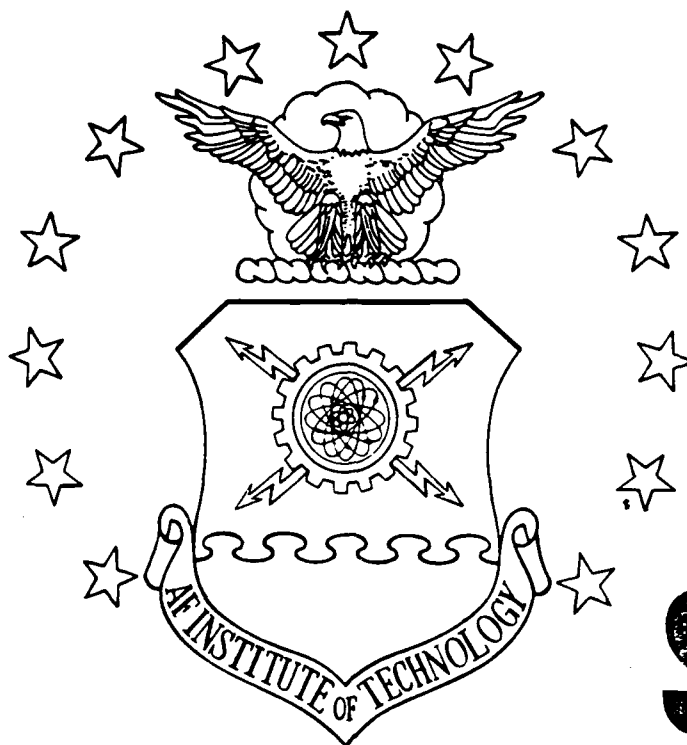
NL





MICROCOPY RESOLUTION TEST CHART
 NBS 1963-A

AD-A194 625



DTIC

ESTATE

JUN 23 1988

S

D

20H

LRS-II: A SPECIALIZED KNOWLEDGE SYSTEM
FOR LAUNCH RESOURCE SCHEDULING

THESIS

James E. Crawford Jr.
Captain, USAF

AFIT/GE/ENG/87D-11

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

88 6 23 046

AFIT/GE/ENG/87D-11

LRS-II: A SPECIALIZED KNOWLEDGE SYSTEM
FOR LAUNCH RESOURCE SCHEDULING

THESIS

James E. Crawford Jr.
Captain, USAF

AFIT/GE/ENG/87D-11

DTIC
S E LITE D
JUN 23 1988
G H

Approved for public release; distribution unlimited

AFIT/GE/ENG/87D-11

LRS-II: A SPECIALIZED KNOWLEDGE SYSTEM
FOR LAUNCH RESOURCE SCHEDULING

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

James E. Crawford Jr, B.S.

Captain, USAF

December 1987

Approved for public release; distribution unlimited

Preface

This research developed a specialized knowledge system, called the Launch Resource Scheduling system, which USSPACECOM operators will use as a decision aid to determine if there is sufficient launch capability to meet future satellite requirements and to assess the immediate impact of launch or on-orbit failures.

The success of this research would not have been possible without the support provided by many individuals. I thank my advisor, Lt Col Greg Parnell, for his professional guidance throughout the research effort. I thank Maj Fred Koch for providing an excellent prototype. I greatly appreciate the excellent support provided by Level 5 Research. In particular, Mr. Karl Seiler provided detailed explanations of how Level 5 functions work and provided an unreleased version of the software which greatly enhanced LRS-II operation. I thank USSPACECOM/J30 for sponsoring this research and for twice flying me out to do on-site knowledge engineering. My point of contact, Capt Norm Thompson, provided the knowledge represented within LRS-II and provided insight as to how it would be used operationally; his support was immeasurable. To Capt Mike Rampino, thanks for being a friend and for providing ideas which enhanced LRS-II. Finally, I thank my wife Dolores who has sacrificed greatly to get me through two degrees and for being my personal typist.

Ed Crawford



on For	
A&I	<input checked="checked" type="checkbox"/>
eed	<input type="checkbox"/>
ation	<input type="checkbox"/>
ation/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Table of Contents

	Page
Preface	ii
List of Figures	v
List of Tables	vi
Abstract	vii
I. Introduction	1
Background	1
Problem	4
Objective	4
Scope	5
Assumptions	7
Materials and Equipment	7
II. Methodology	8
Introduction	8
Understanding the Problem	8
Literature Review	8
Knowledge Engineering	9
System Requirements Identification	9
Development Tool Selection	10
Expert System Development	10
Test and Evaluation	10
User Manual	11
Summary	11
III. Literature Review	12
Introduction	12
Satellite Support Scheduling Advisor	15
Satellite Mission Planner	17
ISIS: Job Shop Scheduling	18
Shuttle Utilization Management System	22
Summary	25
IV. Knowledge Engineering	26
Introduction	26
Launch Manifesting Process	27
LRS Prototype	32
Manifesting Heuristics	33

	Page
V. System Requirements	35
Introduction	35
Input	35
Output	41
Processing	42
Knowledge Base	43
VI. Development Tool Selection	44
Level 5 Description	44
LRS-II Requirements	45
Summary	47
VII. Expert System Development	48
Design Principles	48
Database Formats	49
Manifesting Process	54
Output	58
User Interface	60
Development Iterations	62
VIII. Test and Evaluation	64
Introduction	64
Module Level Testing	64
System Level Testing	66
Field Testing	68
Evaluation	68
IX. Conclusions and Recommendations	70
Summary of Research	70
Recommendations	72
Appendix A: User Manual	75
Bibliography	103
Additional Sources	104
Vita	105

List of Figures

Figure	Page
1. Launch Manifest Process	28
2. Stoplight Output	31
3. Satellite Record	50
4. Launch Vehicle Record	50
5. Launch Pad Record	53
6. Upper Stage Record	53
7. Manifesting Process	55
8. Launch Manifest	59
9. User Interface	61

List of Tables

Table	Page
I. Selected Scheduling Systems	13
II. System Requirements Matrix: Inputs	36
III. System Requirements Matrix: Outputs	37
IV. System Requirements Matrix: Processing	38
V. System Requirements Matrix: Knowledge Base	39
VI. Test Matrix	65
VII. Launch Resource Configurations	67

Abstract

This research used the Level 5 expert system software to develop a specialized knowledge system called the Launch Resource Scheduling system (LRS-II). LRS-II will be used as a decision aid by USSPACECOM to determine if there is sufficient launch capability to meet future satellite requirements and to quickly assess the impact of contingencies such as launch or on-orbit failures.

LRS-II uses multiple knowledge bases to match satellite launch requirements to available launch vehicles, launch pads, and upper stages. Specialized knowledge about satellite requirements and launch resources are stored in dBase III files. Level 5 knowledge base rules match specific fields of the satellite record against fields in the resource records to schedule the earliest launch resources that meet the satellite requirement. If two different types of launch vehicles are capable of launching a satellite, LRS-II selects the launch vehicle that meets the requirement earliest. If the satellite constellation allows multiple satellites on a single launch vehicle, specialized knowledge bases are executed to manifest the additional satellites. During manifesting, the constraints of satellite and resource availability, site processing time, shuttle mission duration, and satellite on-orbit checkout time are used to insure the selected launch date is accurate.

LRS-II Launch Resource Scheduling

I. Introduction

Background

The loss of the space shuttle Challenger and the expendable launch vehicle (ELV) accidents in 1986 and 1987 have created a large backlog of satellites waiting to be launched. This backlog will imperil national security unless the United States effectively manages its launch vehicle resources. These space vehicle losses come at a time when the United States is expanding its satellite coverage.

New satellite constellations to be launched include the Global Positioning System and MILSTAR. The fully operational GPS constellation will consist of 18 active satellites with three satellites in six equally spaced orbital rings and three on-orbit spares. MILSTAR is the Department of Defense's (DOD) most ambitious satellite communication program. It is currently in full scale development and should be fully operational in 1993. The MILSTAR constellation will consist of four satellites in geostationary orbit over the Indian, Atlantic, and eastern and western Pacific oceans, supplemented by three satellites in highly elliptical polar orbits to ensure global coverage (Williams, J., 1987:16-18). These new satellite constellations, as well as replacements to older constellations, require effective management of U.S. launch vehicle resources.

It is the responsibility of U.S. Space Command's Deputy Director for Space Operations (J30) to identify operational needs for current and future space systems. The Control, Launch, and Support Systems Division (J30C) provides staff support to J30 for all matters related to control of space, space launch, and space support systems. J30C is tasked with examining the launch manifests generated by Air Force System Command's (AFSC) Space Division and HQ USAF to determine if there is sufficient launch capability to maintain operational satellite constellations at an acceptable level of performance (Thompson, 1987:1).

In the spring of 1986, J30 requested a computer program be developed to provide an estimate of the launch support required to maintain any number of satellite constellations at a given level of performance. The AFIT thesis research of Major Fred Koch developed a prototype tool which an operator at J30 could use to match satellite requirements against available launch resources (Koch, 1986:2).

The tool developed by Maj Koch is a prototype knowledge based program called the Launch Resource Scheduling System (LRS). LRS begins by having the user enter a start and end schedule day. LRS then selects the highest priority satellite from a satellite database and matches it against a suitable launch vehicle and a compatible pad. If this matching is accomplished before the satellite not later than (NLT) launch day, the satellite is listed as launched and the satellite, launch vehicle, and pad databases are updated. If the NLT day

is exceeded by the time all launch resources are available, the satellite is marked as missing its launch and the next satellite is considered. The scheduling cycle ends when no more satellites are available for launch or the ending schedule day is reached. The LRS output produces a listing which shows each satellite scheduled for launch and the launch resources utilized. The output also lists the launch resources still available at the end of the schedule (Koch, 1986:32-39).

A knowledge based approach to launch resource scheduling was used due to the knowledge intensive nature of the scheduling problem. LRS was implemented using the expert system development tool Insight 2+ on an IBM PC compatible microcomputer. Insight 2+ code is written in IF THEN ELSE production rule format making it easy to understand and easy to modify. Insight 2+ uses dBase format files for storing the required information on satellites, launch vehicles, and launch pads (Koch, 1986:28-30).

J30 was very pleased with the LRS prototype. LRS demonstrated that a knowledge based approach was applicable to the launch manifesting problem. However, the prototype system was limited because it only contained procedural knowledge and lacked specialized knowledge about the satellite constellations under USSPACECOM's operational responsibility. This thesis addresses this major limitation.

Problem

U.S. Space Command staff need a computer program which allows them to do long-range scheduling of launch resources for the satellite constellations under their operational responsibility. These constellations include the Global Positioning System (GPS), the Defense Meteorological Satellite Program (DMSP), the Fleet Satellite Communication System (FLTSAT), the TRANSIT System, and other classified constellations. Also, the program should allow operators to quickly assess the impact of contingencies such as launch or on-orbit failures (Thompson, 1987:2).

Objective

This thesis extends the LRS prototype into a system ready for field testing called the Launch Resource Scheduling System II (LRS-II). USSPACECOM operators can use LRS-II as a decision aid to determine if there is sufficient launch capability to meet future satellite requirements and to quickly assess the impact of contingencies such as launch or on-orbit failures.

Scope

The scope of this thesis reflects the recommendations made by Major Koch and the features desired in an operational system (Boller, 1987). Specific LRS extensions contained in LRS-II include using specialized scheduling knowledge to manifest each satellite constellation; allowing multiple satellites per launch; adding the scheduling of upper stages; allowing the preassigning of launch resources; and adding the constraints of satellite availability, site processing time, shuttle mission duration and refurbishment time, and on-orbit satellite checkout time to the schedule considerations. In addition, LRS-II provides detailed updating of all database files.

The major limitation of LRS was the use of only procedural knowledge to match satellite requirements against available launch resources. The LRS-II system uses specialized knowledge for each satellite constellation to insure the earliest available launch resources are scheduled.

A major consideration not provided in the LRS prototype is launching multiple satellites per launch vehicle. This is a common practice for meeting launch requirements using the space shuttle. If required, the LRS-II system launches multiple satellites per launch vehicle.

Many satellite constellations require upper stages to boost the satellite from a transfer orbit to its final operational orbit. The LRS-II system adds the scheduling of upper stages to the launch manifesting process.

The ability to preassign launch resources is implemented in the LRS-II system. This allows matching satellites to dedicated launch resources, but still allows LRS-II to fit the launch into the schedule where appropriate.

The LRS prototype was limited in the number of launch constraints which were implemented. LRS-II adds satellite availability, site processing time, shuttle mission duration and satellite on-orbit checkout time as constraints on schedule generation. Satellite availability is used to insure a satellite is launched from the correct coast and to determine when the satellite is ready for processing. Site processing time shows the earliest a satellite can be launched once all resources are available. Shuttle mission duration and refurbishment time determines when a shuttle can be reused. Satellite on-orbit checkout time determines when the launched satellite can be used for its operational mission.

LRS-II extensively updates database records when a launch is scheduled. Satellite updating includes recording the earliest day the satellite could be launched; recording the scheduled launch day and the launch resources utilized; recording if resource availability delayed the launch past its desired date and which resources caused the delay; and recording the reason the satellite was not launched if resources are not available. In addition, each resource record is updated to show which satellite utilized it and when it is available again if reusable.

Assumptions

1. It is assumed that satellite requirements are prioritized by earliest desired launch date. J3OC operators will prioritize the requirements using available information.
2. LRS-II must run on a TEMPEST approved microcomputer to allow the manifesting of classified payloads.
3. LRS-II is implemented using the Level 5 expert system development software. J3OC has purchased this software based on the LRS prototype development. Level 5 is the new name for Insight 2+.

Materials and Equipment

The materials and equipment used to implement LRS-II are divided into two categories, computer hardware and software. The computer hardware requirements of the LRS-II system development include a TEMPEST approved microcomputer with at least 640 kilobytes of random access memory (RAM), a 10 megabyte hard disk with a minimum of 2.0 megabytes of free work space, and one double-sided double-density 5 1/4" floppy disk drive. The LRS-II development software used is Level 5, Version 1.0, by Level Five Research, Inc., dBase III, Version 2.41, by Ashton-Tate, Inc., and MS-DOS, Version 2.1 or greater.

II. Methodology

Introduction

The methodology used to conduct this thesis research was based on the systems approach to problem solving. The research problem was divided into components of the overall research task and these component problems were then solved. This chapter describes this systematic approach.

Understanding the Problem

The first task was to gain an overall understanding of J3OC's problem so it could be scoped into an accomplishable thesis research effort. Since no literature was available on how J3OC accomplishes the launch manifesting process, an initial trip to U.S. Space Command was made to confer with the operators at J3OC. The results of this trip are presented in Chapter 4.

Literature Review

With a basic understanding of the problem, the next task was to review applicable literature. Current expert systems in the scheduling domain were reviewed to obtain common factors in design and development tool use. This review is presented in Chapter III.

Knowledge Engineering

Once the literature review was complete, the next task was to acquire the information needed in the knowledge bases. This task is called knowledge engineering and is critical in the development of the expert system. Without thorough and complete knowledge engineering, the expert system would be difficult to design and would not perform properly.

During the knowledge engineering task, the knowledge engineer must learn how the domain experts approach the problem. The knowledge engineer looks for heuristics or rules-of-thumb that the experts use to simplify the search for solutions. The knowledge engineer also establishes the requirements and constraints which will determine how the knowledge is represented in the knowledge base.

Chapter IV covers the knowledge engineering task in detail. It reviews the actual steps taken to acquire the knowledge used in LRS-II. It also explains the current launch manifesting process and how the LRS prototype was implemented.

System Requirements Identification

From knowledge engineering comes the identification of the LRS-II system requirements. This requirements identification includes the necessary inputs and outputs, the processing functions needed, and the information stored in the knowledge base. System requirements are identified in Chapter V.

Development Tool Selection

Once the system requirements were identified, the next task was to insure the Insight 2+ development tool would still meet those requirements. The developer of Insight 2+, Level Five Research, provided a beta copy of their newest version of Insight 2+ which is now called Level 5. Chapter VI presents a summary of the features of Level 5 and explains how it meets the system requirements.

Expert System Development

The next task was the actual development of the LRS-II system. This task included the development of the knowledge bases, databases, and input/output routines required. The expert system development is presented in Chapter VII.

Test and Evaluation

After the initial development of the LRS-II system, the next task was the test and evaluation of the system. Test cases were used to iteratively refine the design of LRS-II. The prototype LRS-II system was field tested at USSPACECOM by J3OC operators to insure it met their requirements and to gain an understanding of the requirements needed in an operational system. A second set of test cases was used to validate the operational LRS-II system and insure it correctly scheduled each satellite constellation. The test and evaluation is presented in Chapter VIII.

User Manual

The final task was to write a user manual. The user manual includes hardcopy of all required software files, presents step by step instructions on how to run the expert system, and explains the output generated by the expert system. In addition, the user manual provides an interactive tutorial which guides the user through a typical consultation. A complete user manual is presented in Appendix A.

Summary

Developing a thorough methodology is crucial to the success of any research project. The methodology used in this research was based on the systems approach. The systems approach logically divided the research into component tasks. The first task was to gain an overall understanding of J3OC's problem. The second task was to complete a thorough literature review of expert systems in the scheduling domain. The third task was to complete the knowledge engineering to acquire the information needed in the expert system knowledge base. The fourth task was to identify the launch manifesting requirements. Using these requirements, the Level 5 development tool was evaluated. The next task was to actually develop the expert system. After the system was developed, test cases were used to refine and validate LRS-II. The last task was to write a good user manual.

III. Literature Review

Introduction

Artificial intelligence applications are becoming more prevalent in both industry and in the military. One area of artificial intelligence is expert systems. Donald Waterman defines expert systems or knowledge-based systems as:

Sophisticated computer programs that manipulate knowledge to solve problems efficiently and effectively in a narrow problem area (Waterman, 1986:xvii).

This thesis research concentrates on using a "specialized knowledge system" to schedule launch resources to meet satellite requirements. A specialized knowledge system is an expert system that contains specialized knowledge about a domain that lacks true experts. To examine common factors in expert system design and development tool use, several scheduling systems were reviewed. These systems are summarized in Table I.

Table I is divided into selected categories which describe each of the expert systems reviewed. System details are written into each column of the table. The first column lists the name of the expert system and the type of scheduling performed. The second column lists the stage of development and the agency where the system was developed. The third column lists the type of software and hardware environments that support the systems.

Table I. Selected Scheduling Systems

NAME/ SCHEDULING TYPE	STAGE OF DEVELOPMENT/ AGENCY	SOFTWARE/ HARDWARE	KNOWLEDGE REP	SCHEDULING ALGORITHM
SATELLITE SUPPORT SCHEDULING SYSTEM	Prototype AFIT	PC Plus PC Scheme IBM PC compatible	Frames Rules	Uses satellite visibility charts to produce PAP schedules for GPS satellites
SATELLITE MISSION PLANNER	Prototype AFIT	PC Scheme Turbo Pascal IBM PC compatible	Frames	Produces schedules which match user requirements by priority against satellite and tracking resources
ISIS Job Shop Scheduling	Prototype Carnegie- Mellon University	Schema Rep Language DEC VAX 11/780	Frames Rules	Uses constraint directed search to schedule plant resources
SHUTTLE UTILIZATION MGMT SYSTEM	Operational Aerospace Corp	Common LISP Symbolics 3640	Rules	Verifies NASA launch manifest and assigns DOD payloads to the NASA manifest

--Continued--

Table I. Selected Scheduling Systems-Continued

NAME/ SCHEDULING TYPE	KERNEL PROBLEM ADDRESSED	EXPERT KNOWLEDGE	INPUT/ OUTPUT	OPTIMAL/ SATISFICE
SATELLITE SUPPORT SCHEDULING SYSTEM	Knowledge Collection Schedule Development	Satellite Control Facility Planner Analysts	Visibility Files Program Action Plans	Satisfice
SATELLITE MISSION PLANNER	Schedule Development	Satellite Control Facility Mission Planners	User Require- ments Mission Schedule	Satisfice
ISIS Job Shop Scheduling	Knowledge Collection Schedule Development Schedule Management	Westing- house Turbine Assembly Plant	Turbine Orders Scheduled Plant Resources	Optimal
SHUTTLE UTILIZATION MGMT SYSTEM	Knowledge Collection Schedule Development	NASA	DOD Payload Requests Launch Manifest	Satisfice

The fourth column states the type of knowledge representation used in the knowledge base. The remaining columns describe some key features of these systems. Each of these systems is presented below.

Satellite Support Scheduling Advisor

In the thesis "A Prototype Expert System Advisor For Satellite Support Scheduling," a prototype satellite support scheduling advisor was developed to demonstrate the applicability of expert system technology to satellite support scheduling at Sunnyvale AFS, CA (Kennedy, 1986).

The scheduling advisor produces visibility charts which show which remote tracking stations can "see" a given satellite at a specific time. The scheduling advisor uses the visibility charts to produce a Program Action Plan (PAP) for each satellite. The PAP is a satellite support schedule which covers a one week period and is used to request range resources. The PAP documents the required satellite supports and provides a time window in which the support must take place. The PAP also identifies support duration times, remote tracking station visibilities during the window, support priorities, and any special information range personnel need to consider when assigning range resources. The prototype system developed produces visibility charts and PAPs which schedule state-of-health supports for the nine Global Positioning System satellites currently in orbit.

There are three key components in the scheduling advisor design. These are the visibility data file for each satellite, the knowledge base, and a dBase II data file which stores the PAP schedule for each satellite.

The visibility data file consists of each satellite's visibility data for an entire week. The file is built as a Lisp function which stores the visibility data in the form of lists.

The knowledge base was built using PC Plus from Texas Instruments. PC Plus, a microcomputer based expert system development tool, was selected because it was backward chaining, supported frames and inheritance, used control rules, supported user defined functions, supported graphics, and interfaced with dBase II.

Kennedy divided the knowledge base into the inputs, the vehicle data, the support requirements, the visibility data, the scheduling rules, and external functions. The external functions are coded in PC Scheme which is a dialect of LISP. The knowledge representation is a combination of frames and rules. The frame structure used by PC Plus can be thought of as being similar to subroutines in a conventional computer language. Each frame performs a subset of the problem with the added advantage of child frames inheriting information from their ancestors. The scheduling advisor was validated using test cases.

Kennedy's research effort culminated with the development of a prototype expert system advisor for scheduling GPS satellite state-of-health supports. The prototype demonstrated the applicability of expert system technology to the PAP scheduling process.

Satellite Mission Planner

In the thesis "A Prototype Knowledge-Based System For Satellite Mission Planning," a prototype satellite mission scheduler was developed to examine the feasibility of using a knowledge-based system to assist mission planners at Sunnyvale AFS in the area of satellite scheduling (Perales, 1986).

The mission scheduler schedules all user requirements by priority unless satellite or tracking resources are unavailable or the desired user requirement conflicts with a higher priority requirement. An example of a user requirement is that the Navy requests a weather satellite to observe the Atlantic Ocean during an exercise. Satellite resources are payload configuration, tape recorders, and power. Tracking resources are the remote tracking stations. The mission scheduler looks at the priority of the requirement and schedules the satellite and tracking resources based on availability of resources.

There are two key components in the mission scheduler: the knowledge base built using PC Scheme and the graphic output produced using Turbo Pascal. Perales selected PC Scheme because it supported frames and inheritance, it supported user defined functions, and it interfaced with external programs.

Perales divided the knowledge base into user requirements, satellite constraints, and tracking station constraints. The knowledge representation used frames with inheritance. PC Scheme does not have a built-in inference strategy, so Capt Perales built a forward chaining control strategy using PC Scheme. The mission scheduler was validated using test cases.

Perale's research effort culminated with the development of a prototype expert system advisor that schedules user requirements by priority against available satellite and tracking station resources. The prototype demonstrated the feasibility of using a knowledge based system to assist satellite mission planners in scheduling user requirements.

ISIS: Job Shop Scheduling

In the dissertation "Constraint-Directed Search: A Case Study of Job-Shop Scheduling," the scheduling system ISIS was presented (Fox, 1983). Fox is currently recognized as a leading expert in the application of expert systems to the scheduling domain (Rowell, 1987). ISIS uses constraint-directed search to solve the job-shop scheduling problem. Fox defines the job-shop scheduling problem as selecting a sequence of operations whose execution results in the completion of an order, and assigning times and resources to each operation. The number of possible schedules grows exponentially with the number of orders, alternative production plans, and possible times to assign resources and perform operations. Fox says that by viewing the scheduling problem

from a constraint-directed search perspective, much of the knowledge can be viewed as constraints on the schedule generation and selection process. The ISIS system constructs schedules which satisfy as many constraints as possible in near real-time.

The key components of ISIS are the modeling system and the constraints. The modeling system was built using the Schema Representation Language (SRL) with constraints attached to the schema of the modeling system. A schema (frame) is a collection of slots and values. Each schema, slot, and/or value may have information attached to it. In addition to attribute knowledge, slots define inter-schema relations through which slots and values may be inherited. The inheritance of a relation is user definable.

The ISIS modeling system is a multi-layer system for modeling manufacturing organizations using SRL. ISIS layers are structural, basic semantics, world semantics, and domain semantics. The basic concepts are those of states, objects, and acts. Acts transform states and objects. Time relations provide ordering among states and acts. A manufacturing operation is defined as an act, and time and causality relations link it to other states and acts. Prototypes, instances, and manifestations are used to distinguish classes, elements, and states of objects and acts. Allocation of resources is defined as a state of possession by some act.

Examples of scheduling terminology are defined below. An operation schema is a prototype with the single attribute work-center. A reservation allocates resources to scheduled operations. A product is a type of object produced by an operation. An order is a sub-type of the shipped state which defines the product, quantity, customer, and ship date.

The four layers of ISIS provide primitives for describing the common actions and states in the scheduling domain. ISIS is extended by providing the capability to attach constraints to its schema, slots, and values. Constraints are not always satisfiable, so relaxations are also added to ISIS. Relaxations allow selection of alternatives to a specified constraint. Not all constraints are equal. ISIS specifies the importance of a constraint by allowing assignment of a value between 0 and 1. Obligations define conditions under which a constraint must be considered and possibly satisfied. Factors that affect the obligation are: time over which the constraint is applicable, the consistency of the constraint, the source of the constraint, and the context of the constraint. Actual constraint generation occurs dynamically by attaching constraint generators to relations in ISIS.

Fox says the goal of ISIS is to construct schedules which satisfy as many constraints as possible in near real-time (Fox, 1983:11). ISIS performs a hierarchical, constraint-directed search to construct schedules. Constraint-based pre and post-search analysis are used to bound the solution space before performing search, and help diagnose poor constraint

decisions at other levels. Constraints are dynamically generated and evaluated during the search process.

Hierarchically constrained search communicates constraints from a higher level to a lower level in order to guide search at the next level.

Fox believes that ISIS's approach to scheduling utilizes an architecture which is becoming a standard in AI in systems like Hearsay-II, SU/x, SU/p, and Molgen (Fox, 1983:85). However, Fox feels what sets ISIS's approach apart is that decisions at a higher level do not restrict decisions at a lower level, but constrain them. The constraints guide the search, but may be relaxed if they prove overly restrictive.

In summary, Fox's contributions are in three areas: representation, job shop scheduling, and constraint-directed search. Representation provides a more complete semantics for the modeling of organizations which includes states, acts, time, causality, multi-level representation, and support for discrete simulation.

The contribution to job shop scheduling is a system which can represent and consider all the domain constraints during the construction of a schedule. It also provides incremental scheduling in reaction to changes in the plant's status, and suggests alternative schedules when constraints cannot be satisfied.

Shuttle Utilization Management System

In the report "SUMS-A Shuttle Utilization Management System," an operational payload manifesting system was developed for the Utilization Planning Directorate (UPD) of the Aerospace Corporation (Foonberg, 1986). UPD is responsible for advising the Air Force on the feasibility of NASA's shuttle launch manifests. These manifests report the scheduled launch dates of shuttle missions for the next several years as well as which payloads are to fly on each mission. It is also the job of UPD to schedule DOD payloads on these missions.

The SUMS system incorporates the rules and constraints of the Space Transportation System (STS) in order to analyze STS manifests and schedule DOD payloads on the DOD missions. There are three parts of SUMS: the examination of interval constraints, the scheduling of DOD payloads, and the assignment of airborne support equipment.

The software which checks for violations of interval constraints resembles OPS5. A database of rules describes the constraints, and code does the actual checking. Each rule is designed to look at two flights and determine if there is enough time between those flights. If the required number of days indicated by the rule is not met, a violation of the interval is reported to the user. If the user requests, SUMS can fix the violations as well as explain them.

The software which schedules DOD payloads is divided into two parts. First, the file containing the DOD requests for payloads must be parsed and stored in the SUMS database. Second, the planning phase assigns the requests among the DOD assigned slots.

The request file is represented as LISP forms. Each LISP form is a list of properties of the payload. The elements of each list are program name, initial operational capability, expected launch date, launch site, upper stage, launch on need or launch on schedule, and priority.

The planning phase does the actual assignment of payloads. Each DOD slot is processed one at a time. For each slot, a primary payload and up to two backup payloads are scheduled. SUMS determines which payloads are available, orders those payloads by priority, and then makes the actual payload assignments. SUMS uses a rulebase to implement the scheduling process.

The last part of SUMS does the assignment of critical airborne support equipment (ASE) to those payloads requiring it. The ASE serves as a cradle to support and cool PAM-D upperstages. SUMS collects the PAM-D payloads from the launch manifest and assigns the ASE based on priority of payloads. Any payloads not receiving ASE are rescheduled.

The knowledge base of SUMS was built using Common LISP. Foonberg selected Common LISP because he wanted a language which allowed rapid prototyping and the possibility of a rule-based system, and Aerospace did not have any expert system

development tools (Foonberg, 1987). OPS5 was considered, but it was not chosen due to its inability to express certain concepts necessary in SUMS.

The hardware SUMS was designed to run on is the Symbolics 3640 LISP Machine. A version of SUMS ran on a VAX 11/780, but it proved too slow. A particular task that took 30 minutes on a loaded VAX only required six seconds on the 3640.

Foonberg says SUMS is constantly evolving. Recently, the knowledge base was modified to include NASA and DOD manifesting rules. In addition, the user now has the capability to backtrack through previous decisions to remove bottlenecks. Also, a version of SUMS implemented in C runs on a Compaq personal computer; this allows the scheduling of classified payloads (Foonberg, 1987).

Summary

The scheduling systems reviewed presented several key concepts that were applied to the LRS-II development. Each of the systems used either frames or knowledge bases to partition the knowledge into logical functions. Constraints are a major factor in schedule generation and constraint relaxation may be necessary to create a schedule. Scheduling conflicts occur and methods must be created to handle these conflicts. The software environment can be a bare knowledge representation language such as PC Scheme or SRL, where the knowledge engineer specifies the control and interface; or the software environment can be an expert system development tool such as PC Plus, where the control and interfaces are specified. With an understanding of current scheduling systems, the next task was to complete knowledge engineering. Knowledge engineering is presented in Chapter IV.

IV. Knowledge Engineering

Introduction

The knowledge engineering (KE) phase of the research forms the foundation of the expert system development process. From direct interaction with J3OC came the system requirements and the rules used in LRS-II. The KE phase of the research was conducted at USSPACECOM during two one-week trips. The first trip to USSPACECOM was used to gain an overall understanding of J3OC's problem so it could be scoped into an accomplishable thesis research effort. The second trip was used to demonstrate the LRS-II prototype and to refine the prototype requirements to those needed in an operational system. The primary J3OC contact was Captain Norm Thompson, Space Launch Systems Operations Officer.

During my USSPACECOM visits, we discussed the launch manifesting process, the heuristics used to match satellites to launch resources, and the requirements that should be implemented in the LRS-II system. In addition, the implementation of the LRS prototype and its limitations were explained by its developer, Major Koch. Each of these areas is presented next.

Launch Manifesting Process

The launch manifest process is shown in Figure 1 (Dutry, 1987). Space Division prepares a draft DOD mission model based on System Program Office (SPO) launch requirements, available ELVs, and the NASA space shuttle manifest. The Space Division mission model goes to AFSC for approval. Then, HQ USAF chairs the DOD Space Launch Users Committee to confirm Service support for POM requirements to pay for the DOD missions. The DOD mission model is then reconciled against the available ELV assets and the required STS launch capability is negotiated with NASA.

The existing launch manifest process has evolved over a period of years. Since the formation of USSPACECOM in 1985, they have participated in the manifesting process as an observer. J3OC was tasked with reviewing the manifests generated by Space Division and HQ USAF to insure there was sufficient launch capability to maintain operational satellite constellations at an acceptable level of performance. This task was relatively simple prior to the Challenger loss, because all DOD payloads were primarily manifested for the space shuttle, although a few ELVs were used.

However, the loss of the Challenger has led to the mixed fleet concept where ELVs and space shuttle vehicles are both used to meet DOD launch requirements. This has led to a proliferation of ELVs including the Titan 4, Titan 2, Delta 2 Medium Launch Vehicle (MLV), and the planned Heavy Lift Vehicle (HLV).

```

SPOs ----->:
:
: -----
Space Test ->: SD / YO : <-- NASA
:
: -----
Other ----->: : <-- SD/YX (ELV Schedule)
:
:
: --DRAFT
:
: -----
: SD / CC :
: -----
:
: --SUPPORT VERIFIED
:
: -----
: Staffing :
: By DOD :
: -----
:
: --SEQUENCE VERIFIED
:
:
: -----
USAF ----->: : <-- OJCS
: : DOD Space : :
USA ----->: : Launch : <-- OSD
: ----->: Users : <--
USN ----->: : Committee : <-- DDR&E
: :
SDIO ----->: : <-- SAF/SS
:
:
: --SEQUENCE APPROVED
:
: -----
: SD / CC :
: NASA :
: -----
:
: --APPROVED MANIFEST
:
: -----
: Budget :
: Submittal :
:
: NASA :
: Schedule :
: -----

```

28

In 1988 USSPACECOM becomes a voting member of the DOD Space Launch Users Committee and will directly advocate the requirements of its component commands (AFSPACECOM, NAVSPACECOM, USASA) and the other unified and specified commands. The proliferation of launch vehicles and increasing satellite requirements led J3OC to request a computer program to assist them in matching satellite requirements against launch resources. Lt Col Boller explained that they need a computer program which allows them to build an 8 year launch manifest and also allows them to ask "what-if" questions. "The program should serve as a long range scheduler and it should also allow us to assess day to day impacts such as the loss of a satellite" (Boller, 1987). USSPACECOM plans to use LRS-II to build launch manifests for the satellite constellations for which they have operational responsibility.

USSPACECOM uses the STOPLIGHT computer program to determine the status of their on-orbit satellite constellations and to predict when satellites must be launched to keep the constellations operational. STOPLIGHT is a microcomputer based computer program developed by AFSPACECOM/DOS. STOPLIGHT provides decision-makers with the ability to quickly review current and predicted on-orbit satellite capability and to evaluate proposed launch manifests (Williams, T., 1987).

The STOPLIGHT output gives the predicted status of the satellite constellation for the next eight years. Figure 2 is the output of STOPLIGHT for the GPS satellite constellation. STOPLIGHT uses the projected end of life (EOL) of each satellite to predict the system status. A ratio, of the number of predicted healthy satellites to the number of required operational satellites, determines the predicted system status. If the ratio is 90% or more, the status is green. If the ratio is between 67% and 89% (more than 2/3), the status is yellow. If the ratio is less than 67% (less than 2/3), the status is red; hence the name STOPLIGHT.

User requirements as reflected in STOPLIGHT need dates tell USSPACECOM when a launch is required to keep the satellite constellation at its full on-orbit requirement. However, USSPACECOM does not have any program which matches satellite requirements to available launch resources. This is how they plan to use LRS-II.

The plan is that a USSPACECOM operator will take the latest STOPLIGHT output and use it to determine which satellites require launch. The operator will then prioritize the satellite requirements by earliest desired launch date. The operator then determines which launch resources (upperstages, launch vehicles, pads) are available and LRS-II will match the satellite requirements against the available launch resources to build a launch manifest.

04-23-1987

Classification: UNCLASSIFIED

DETAIL SYSTEM STATUS

MISSION: Navigation PROGRAM: NAVSTAR OPSCAP: N/A (R&D)
 Constellation: Required: 18 / 3 On-Orbit: 6 / 0 Checkout 1 Months
 Booster/Launch Complex: Atlas/WSMC SLC-3 (STS and Delta/ESMC)

SPACECRAFT	GPS-3	GPS-4	GPS-5	GPS-6	GPS-10	GPS-11
Launch Date	10 / 78	4 / 80	7 / 83	6 / 84	9 / 84	10 / 85
Proj EOL	3 / 87	4 / 88	7 / 91	6 / 92	9 / 92	10 / 93

EOL Source: SD GAP DATA

CY	1987	1988	1989	1990	1991	1992	1993	1994
Need Dates	1	1						
Next Available	A A A A A A	A A A A A A	A A A A A A	A A A A A A				
Ground System				A User Set Production				
Launch Schedule		A A A A A A A A	A A A A A A A A	A A A A A A A A				
Predicted System Status	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	YYYYYY566666	666666666666	666666666666	666666666666	666666666666

Comments: Deltas available Oct 89. Although R&D, NAVSTAR Block 1 supports Trident testing and other users

Classification: UNCLASSIFIED

AFSPACECOM/DOS

Figure 2. Stoplight Output

The manifest generated by LRS-II will be used as input to STOPLIGHT to determine the new predicted status of each constellation. The USSPACECOM operator can iteratively change the numbers and types of launch resources available and use the manifests generated by LRS-II as inputs to STOPLIGHT. This provides the USSPACECOM operator with the capability to assess the impact of using different launch resource configurations to meet satellite requirements. In addition, if resource capability changes for a particular satellite constellation, the operator can easily modify the appropriate knowledge base without affecting the rest of the system. This allows USSPACECOM operators to assess the impact of future launch systems and to plan for the right mix of launch resources.

LRS Prototype

The LRS prototype demonstrated that a knowledge based approach was appropriate for the resource scheduling problem (Koch, 1987:72). LRS used a single knowledge base and a set of procedural rules to match satellite requirements against available launch resources. The LRS scheduling process begins with the user entering the initial and end schedule day. The knowledge base then looks for an available satellite with the highest priority by searching a database containing satellite requirements, available launch vehicles, and available launch pads. Once a satellite is selected, the next step is to find a suitable launch vehicle by matching the launch vehicle field of the satellite record against available launch vehicles. If no

launch vehicle is available, the satellite is marked as missing its launch. If a launch vehicle is available, the same matching process is followed for selecting a launch pad. If the satellite NLT date is exceeded before resources are available the satellite is marked as missing its launch. The scheduling process continues until all satellite requirements are processed. The output of LRS is a launch manifest, a list of unscheduled satellites, and a list of available resources at the end of the scheduling period.

Manifesting Heuristics

The major weakness of the LRS prototype was that it only used procedural rules to manifest satellites. The highest priority satellite was matched to a suitable launch vehicle and a compatible pad. However, there was no specialized knowledge detailing how to match specific launch resources to specific satellites. For example, in the operational world, a GPS satellite scheduled on a Delta flies alone, but GPS satellites scheduled for the shuttle launch in pairs. Or, a Nova satellite flies alone on a Scout while Oscar satellites fly in pairs on a Scout. Due to the inability of LRS to manifest specific USSPACECOM satellites, it has not been used since its delivery in 1986.

The heuristics of matching specific satellites to specific launch resources requires specialized knowledge for each satellite constellation. This insures the correct matching and allows for ease of maintenance of the knowledge base when

constellations are added or deleted. The heuristics for manifesting satellites USSPACECOM has operational responsibility for were obtained by interviewing Captain Thompson. These heuristics form the basis of LRS-II.

Other features requested by J30C in the LRS-II system that were not implemented in LRS are allowing multiple satellites per launch, adding the scheduling of upper stages for those satellites requiring them, allowing the preassigning of launch resources, and adding the constraints of satellite availability, site processing time, shuttle mission duration, and satellite on-orbit checkout time to the schedule considerations. These features are explained in the scope section of Chapter I.

With an understanding of the launch manifesting process and the limitations of the LRS prototype, the actual LRS-II system requirements were developed. These system requirements are presented in the next chapter.

V. System Requirements

Introduction

The requirements for LRS-II can be divided into four main areas: the inputs, outputs, processing, and the knowledge base. The development of LRS-II was to initially build the prototype system which could be tested at USSPACECOM and then to iteratively incorporate the requirements of the operational system. System requirements matrices are presented in Tables II, III, IV, and V. Each table outlines specific requirements for the prototype and operational system.

Input

Table II contains the system input requirements. All system inputs are stored external to the knowledge base in dBase files except for the manifesting period which is entered interactively by the user.

The satellite desired launch date comes from the output of the STOPLIGHT program. The USSPACECOM operator uses the the STOPLIGHT need dates to prioritize the satellite requirements by earliest desired launch date and enters these requirements into the satellite database. Specialized knowledge about each satellite including required launch vehicles, required upper stages, and other launch constraint data is also stored in the satellite database.

Table II.

SYSTEM REQUIREMENTS MATRIX: INPUTS

REQUIREMENTS	PROTOTYPE SYSTEM	OPERATIONAL SYSTEM
SATELLITE REQUIREMENTS	5 CONSTELLATIONS 10 LAUNCHES	13 CONSTELLATIONS 20 LAUNCHES
UPPER STAGE RESOURCES	3 TYPES	3 TYPES
LAUNCH VEHICLE RESOURCES	5 TYPES 3 SHUTTLES	7 TYPES 4 SHUTTLES
LAUNCH PAD RESOURCES	7 PADS	10 PADS
MANIFESTING PERIOD	INTERACTIVE ENTRY	INTERACTIVE ENTRY

NOTE : ALL INPUT DATA IS STORED IN DBASE III FILES
EXCEPT MANIFESTING PERIOD

Table III.

SYSTEM REQUIREMENTS MATRIX: OUTPUTS

REQUIREMENTS	PROTOTYPE SYSTEM	OPERATIONAL SYSTEM
LAUNCH MANIFEST LISTING	HARDCOPY LISTING SATELLITE SCHEDULED DESIRED LAUNCH DATE ACTUAL LAUNCH DATE LAUNCH VEHICLE LAUNCH PAD UPPER STAGE	ADDITIONALLY EARLIEST LAUNCH IOC REASON MISSED SCHEDULE CHOICE
UNSATISFIED SATELLITE REQUIREMENT LISTING	ENTRY IN THE LAUNCH MANIFEST SHOWING A ZERO VALUE FOR THE ACTUAL LAUNCH DATE	LISTS SATELLITE MISSING LAUNCH REASON MISSED - NO VEHICLE - NO PAD - NO STAGE - NLT EXCEEDED
AVAILABLE RESOURCE LISTING	LISTS THE UNUSED RESOURCES AND WHEN THEY ARE AVAILABLE	SAME

NOTE : ALL OUTPUT LISTINGS CAN BE DISPLAYED, PRINTED, AND SAVED

Table IV.

SYSTEM REQUIREMENTS MATRIX: PROCESSING

REQUIREMENTS	PROTOTYPE SYSTEM	OPERATIONAL SYSTEM
SELECT A SATELLITE TO SCHEDULE	EARLIEST DESIRED LAUNCH DATE	SKIP OVER LAUNCHED SATELLITES PROCESS SATELLITES THAT MISSED LAUNCH IN EARLIER RUNS
LAUNCH VEHICLE MATCH	LIMITED HEURISTICS	BUILDS PATHS USING TWO TYPES OF LAUNCH VEHICLES AND SELECTS THE SHORTEST PATH TO LAUNCH SATELLITE
LAUNCH PAD MATCH	LIMITED HEURISTICS	MATCH PAD TO SELECTED VEHICLE AND COMPUTE SITE PROCESSING TIME FROM DAY PAD AVAILABLE
UPPER STAGE MATCH	LIMITED HEURISTICS	INSURE STAGE IS COMPATIBLE WITH SATELLITE, LAUNCH VEHICLE AND PAD
UPDATE DBASE FILES	LIMITED UPDATING	DETAILED UPDATING OF SATELLITE FILE RESOURCE FILES SHOW HOW RESOURCE WAS UTILIZED

Table V.

SYSTEM REQUIREMENTS MATRIX: KNOWLEDGE BASE

REQUIREMENTS	PROTOTYPE SYSTEM	OPERATIONAL SYSTEM
MANIFESTING RULES	SINGLE KNOWLEDGE BASE CONTAINS ALL MANIFESTING RULES	SEPARATE KNOWLEDGE BASES FOR EACH SATELLITE CONSTELLATION AS NEEDED
USER INTERFACE	SEPARATE KNOWLEDGE BASE ALLOWS USER TO MODIFY DATABASES AND START MANIFESTING PROCESS	USER CAN SPECIFY SINGLE OR MULTIPLE SATELLITES PER LAUNCH VEHICLE USER CAN INTER- ACTIVELY MATCH LAUNCH RESOURCES TO SATELLITES
INPUT / OUTPUT HANDLING	PASCAL ROUTINES RETRIEVE INPUT FROM THE DBASE FILES AND CREATE OUTPUT LISTINGS	BUILT-IN FUNCTIONS HANDLE I/O OUTPUT LISTINGS CREATED BY LEVEL 5

In the prototype system, five of the constellations USSPACECOM has operational responsibility for were scheduled. In the operational system, eight additional constellations are scheduled.

There are three types of upper stage resources. These are the GPS payload assist module (SGS), the inertial upper stage (IUS), and the Centaur vehicle. These upper stage vehicles are used to boost the satellite from a transfer orbit to the final orbit.

There were five types of launch vehicle resources in the prototype system. These are the Delta MLV, Scout, Atlas E, Atlas G, and Space Shuttle. In addition, the operational system includes the Titan 2 and Titan 4 launch vehicles.

There are ten types of launch pad resources in the operational system. These launch pad resources are on the east coast and west coast. Generally, each satellite must be launched off of a specific coast on a specific launch pad.

The only input entered interactively into the LRS-II system is the manifesting period. The USSPACECOM operator enters a begin and end schedule day. The user also has the option of only entering a begin schedule day and allowing LRS-II to process all satellite requirements.

Output

Table III contains the system output requirements. The outputs produced by LRS-II are the launch manifest, the list of unsatisfied satellite requirements, and the list of available launch resources. All outputs are hardcopy listings.

The launch manifest is a complete listing of the satellites scheduled for launch and the launch resources scheduled. The listing shows the satellite scheduled, the desired launch date, the earliest launch date, the actual launch date, the satellite initial operational capability (IOC) date, the launch vehicle scheduled, the launch pad scheduled, and the upper stage scheduled to support the launch. The listing also shows the scheduling choice of singly or multiply for satellite constellations allowing multiple satellites on a single launch.

The list of unsatisfied satellite requirements lists the satellites missing launch and the reason the requirement was not met. These reasons are a lack of launch resources or the satellite NLT launch day exceeded before the satellite was scheduled for launch.

The last output is a listing of the launch resources available at the end of the manifesting period. This listing provides the USSPACECOM operator with a means of analyzing the utilization of launch resources.

Processing

Table IV contains the system processing requirements. These requirements list the basic steps involved within the LRS-II system to create a launch manifest. The process begins with LRS-II selecting the satellite with the earliest desired launch date from the satellite database. LRS-II then schedules launch resources using the scheduling heuristics applicable to the particular satellite constellation.

When an upper stage, launch vehicle, and launch pad have been selected, the dBase files for each launch resource and the satellite requirement are updated. The process then continues with the next satellite considered until all satellites are processed.

The prototype system uses limited heuristics in manifesting satellites. The operational system uses more constraints in matching satellite requirements against launch resources. In addition, the user has the capability to interactively enter data during each processing phase.

Knowledge Base

Table V contains the knowledge base requirements. The manifesting rules are stored in a central knowledge base. This knowledge base attempts to process each satellite requirement and passes control to specialized knowledge bases to handle particular satellite requirements as needed. A separate knowledge base controls all interaction with the user. This knowledge base allows the user to reset and view database files, and to start the manifesting process.

In the basic system, input/output routines were written in Pascal. Insight 2+ directly supports the Pascal language to allow interfacing with dBase files. This allows the knowledge base to retrieve data from the dBase file, process the data using heuristics, update dBase files as necessary, and produce the desired output listings. The extended system uses built-in Level 5 functions to allow direct database access.

VI. DEVELOPMENT TOOL SELECTION

Level 5 Description

Level 5 (L5) is an advanced environment for expert system development using microcomputers. L5 uses both backward chaining goal driven inference and forward chaining. During backward chaining, L5 pursues a specific goal, searching for conditions that support that goal. During forward chaining, L5 matches contextual data to a pattern or template described by the rules of the knowledge base.

The knowledge representation language is Production Rule Language (PRL). In PRL knowledge is represented as IF...AND...THEN...ELSE English language rules, which contain the factual data comprising the domain of the expert system. PRL also allows the user to specify procedural rules, manipulate numeric data, and to attach confidence levels. Knowledge bases created by L5 are run in compiled form, allowing fast execution.

L5 knowledge bases may chain to other knowledge bases and communicate with them through global facts. This allows partitioning the knowledge as appropriate. External programs can be activated directly by a knowledge base allowing access to Pascal, C, and other conventional programming languages, as well as special software packages. L5 also allows direct access to dBase database files using built-in functions.

L5 contains a menu-driven user interface which allows selection of a window-based editor for the creation of knowledge bases, allows the compilation of created knowledge bases, and allows direct access of up to three external programs.

The L5 inference engine processes the compiled knowledge base making queries of the user, executing external programs, and evaluating the rules of the knowledge base to reach conclusions. The knowledge engineer may specify the format of queries presented to the user. For each query, the user may be asked to select from a menu or to enter textual data.

Anytime during a consultation, the user can request all of the system functions or pertinent help screens. The user can select from several report types to explain the current line of reasoning. Reports include a full trace of all rules fired, the current rule chain, all the rules which reach the same conclusion, any rule or fact in the knowledge base, all conclusions reached, and all facts and values obtained or inferred.

LRS-II Requirements

The Level 5 software development environment provided an excellent match to LRS-II system requirements. The system inputs include satellite requirements, upper stage resources, launch vehicle resources, and launch pad resources. Each requirement or resource is stored in a database record. L5 allows direct creation and modification of databases by

allowing the installation of dBase on the main menu and using built-in functions to allow direct database access when LRS-II is executing. The manifesting period is obtained through direct query of the user.

The system outputs are hardcopy listings of the launch manifest, the unsatisfied requirements, and the available launch resources. L5 allows the creation of any format for listings and allows the inclusion of current contextual data in these listings.

The system processing requirements include the steps involved within the LRS-II system to create a launch manifest. These steps are represented in L5 as PRL rules. PRL English rules make LRS-II very readable and maintainable for the user. The L5 inference engine uses backward chaining to pursue the goal of manifesting satellites by matching launch vehicles, launch pads, and upper stage resources against satellite requirements. The L5 editor allows easy creation of knowledge bases and provides direct compilation. Errors encountered during compilation are listed and L5 reenters editing at the point where the error occurred.

The knowledge base requirements are the static knowledge that remains unchanged during LRS-II processing. Separate knowledge bases for satellite constellations are easily created and used in the L5 environment. Chaining allows one knowledge base to pass control to another knowledge base when called. The L5 reporting system allows excellent debugging of a created knowledge base by allowing access to the line of reasoning, the

current rule chain, or any fact or rule. The reporting system also can be used to modify facts during LRS-II processing. This allows direct interaction by the user during processing. The LRS-II user interface was easily created using the menu creation capability of L5. A separate knowledge base allows the user to reset and view databases, and to start the manifesting process. The L5 database functions eliminate the need for Pascal routines to handle input/output and increase processing speed greatly.

Summary

The Level 5 software development environment provided an excellent match to the LRS-II system requirements. The L5 features include the PRL English rules, built-in database access functions, external program activation, knowledge base chaining, and excellent debugging aids. With the development tool matched to the system requirements, the LRS-II system development could begin.

VII. Expert System Development

Design Principles

Expert system design is the critical step in the development process. This step matches the system requirements to the development tool. The primary development goal was to design and build the prototype system as outlined in the System Requirements, and then to iteratively modify the design to incorporate the operational system. To achieve this development goal, several software design principles were adhered to.

First, the system design was kept generic for all satellite constellations. This allowed easy extension of the basic system from a five constellation system to a thirteen constellation system.

Second, the system design maintained modularity to allow ease of testing and maintainability. A modular system allows each piece to be individually tested and then the entire system to be tested as an integrated package. This form of testing increases the probability the system will operate correctly. Once the system is fielded, it must be maintained by the user. Modularity allows the user to modify a part of the system without affecting the rest of the system.

Third, the design and construction of the system should represent the logical flow of information as understood by the user. To achieve this, USSPACECOM was asked how they match

satellite requirements against launch resources. Their logic forms the heart of the manifesting rulebase. Also, USSPACECOM was asked what type of information should be reflected in the database records and the output generated by LRS-II. Each database and output listing displays information in the format they want to see.

Lastly, the system should be easy to use. A menu-driven interface allows the user to modify databases, print output listings, and start schedule generation. A detailed user manual (Atch A) provides instructions on how to run LRS-II and how to maintain it as modifications become necessary.

The rest of this chapter presents the satellite requirement and launch resource database record formats, explains how satellite requirements are matched to launch resources, shows examples of the output listings generated, and discusses the user interface.

Database Formats

An example of a satellite requirement record is shown in Figure 3. The Satellite Name and Constellation uniquely identify each satellite. Launch Vehicle and Upper Stage are used to identify the types of resources the particular satellite requires. Coast identifies whether the satellite must be launched from the east coast or west coast.

Information for Satellite Record Number 4

Satellite Name	GPS-4
Constellation	GPS
Launch Vehicle 1	SHUTTLE
Launch Vehicle 2	DELTA
Upper Stage 1	SGS
Upper Stage 2	NONE
Coast	EAST
Available	4
Site Processing Time	14
Desired Launch Date	4
Not Later Than Launch Date	999999
On-Orbit Checkout Time	30
Launched	L
Earliest Launch Date	132
Launch Date	132
Initial Operational Capability	162
Vehicle Used	OV-1
Pad Used	SLC-39A
Stage Used	SGS-4
Vehicle Delay	Y
Pad Delay	Y
Stage Delay	Y
Reason Satellite Not Launched1	NONE
Reason Satellite Not Launched2	NONE

Figure 3. Satellite Record

Information for Vehicle Record Number 3

Vehicle Name	OV-1
Vehicle Type	SHUTTLE
East Pad 1	SLC-39A
East Pad 2	SLC-39B
West Pad 1	NONE1
West Pad 2	NONE1
Upper Stage 1	SGS
Upper Stage 2	IUS
First Available	30
Site Processing Time	60
Mission Duration	14
Turn Time	90
Launch Date	132
First Satellite	GPS-3
Second Satellite	GPS-4
Next Available	236

Figure 4. Launch Vehicle Record

The next five fields specify time related information about the satellite: when the satellite is available, how long it takes to process at the launch pad, when the satellite launch is desired, when a launch is too late, and how long it takes to checkout the on-orbit satellite.

The remaining fields are modified when the satellite is processed by LRS-II. Launch is marked with a "L" when the satellite is manifested and a "X" when the satellite misses launch. Launch Date and Initial Operational Capability identify when the manifested launch occurs and how long after launch before the satellite is operational. The next six fields specify the actual resources used and whether availability of the resource caused a delay in meeting the satellite Desired Launch Date. The last two fields specify why the satellite could not be scheduled using the launch paths generated by either Launch Vehicle 1 or Launch Vehicle 2. These reasons include launch vehicle, launch pad, or upper stage unavailable, or the satellite NLT day exceeded.

An example of a launch vehicle record is shown in Figure 4. The Vehicle Name and Vehicle Type uniquely identify each launch vehicle. The pad and upper stage fields identify the specific types used by the launch vehicle.

The next four fields specify time related information about the launch vehicle: when the launch vehicle is first available, how long it takes to process at the launch pad, how long the launch vehicle is in space, and how long it takes to refurbish once it lands. The two primary classes of launch vehicles are ELV's and reusable shuttles. The Mission Duration and vehicle Turn Time only apply to shuttle vehicles.

The remaining fields are modified if the launch vehicle is actually used. First Satellite and Second Satellite identify the satellites manifested. Next Available identifies when reusable shuttles can be used again.

An example of a launch pad record is shown in Figure 5. The Pad Name, Pad Type, Coast, and Upper Stage uniquely identify the pad, what type of launch vehicle it processes, where the pad is located, and the type of upperstage the pad can process. Generally, a given pad can only launch a single type of launch vehicle. Pad Turn Time specifies how long after a launch before the pad can be used again.

An example of an upper stage record is shown in Figure 6. Stage Name and Stage Type uniquely identify each upper stage. Launch Date and Satellite Boosted specify when the upper stage was used and which satellite was boosted. The example database records show the result of updating after a GPS-4 satellite was manifested. These records should be referred to during the explanation of the manifesting process.

Information for Pad Record Number 3

Pad Name	SLC-39A
Pad Type	SHUTTLE
Coast	EAST
Upper Stage 1	IUS
Upper Stage 2	SGS
First Available	30
Turn Time	30
Next Available	162

Figure 5. Launch Pad Record

Information for Stage Record Number 4

Stage Name	SGS-4
Stage Type	SGS
First Available	40
Site Processing Time	7
Launch Date	132
Satellite Boosted	GPS-4

Figure 6. Upper Stage Record

Manifesting Process

LRS-II uses four knowledge bases to allow complete manifesting of all 13 constellations. However, a single knowledge base does all scheduling of a single satellite to a launch vehicle. This manifesting process is shown in Figure 7.

Processing begins with LRS-II selecting the satellite requirement with the earliest Desired Launch Date from the satellite database (the database must be ordered). If the satellite is marked as Launched, LRS-II selects the next satellite until it finds an unsatisfied satellite requirement.

The next step is to match the earliest available launch vehicle to the satellite. This is done by matching the Launch Vehicle 1 field of the satellite record against the Vehicle Type field of each launch vehicle record until the earliest available type 1 launch vehicle is found. The Upper Stage fields of the satellite and launch vehicle are also matched to insure the selected vehicle can accommodate the required upper stage.

Next, the earliest available launch pad is matched to the selected type 1 launch vehicle. This is done by matching the Pad fields of the launch vehicle record against the Pad Type field of each launch pad record until the earliest available launch pad is found. The Coast field of the satellite record and the pad record are also matched to insure a pad on the correct coast is selected.

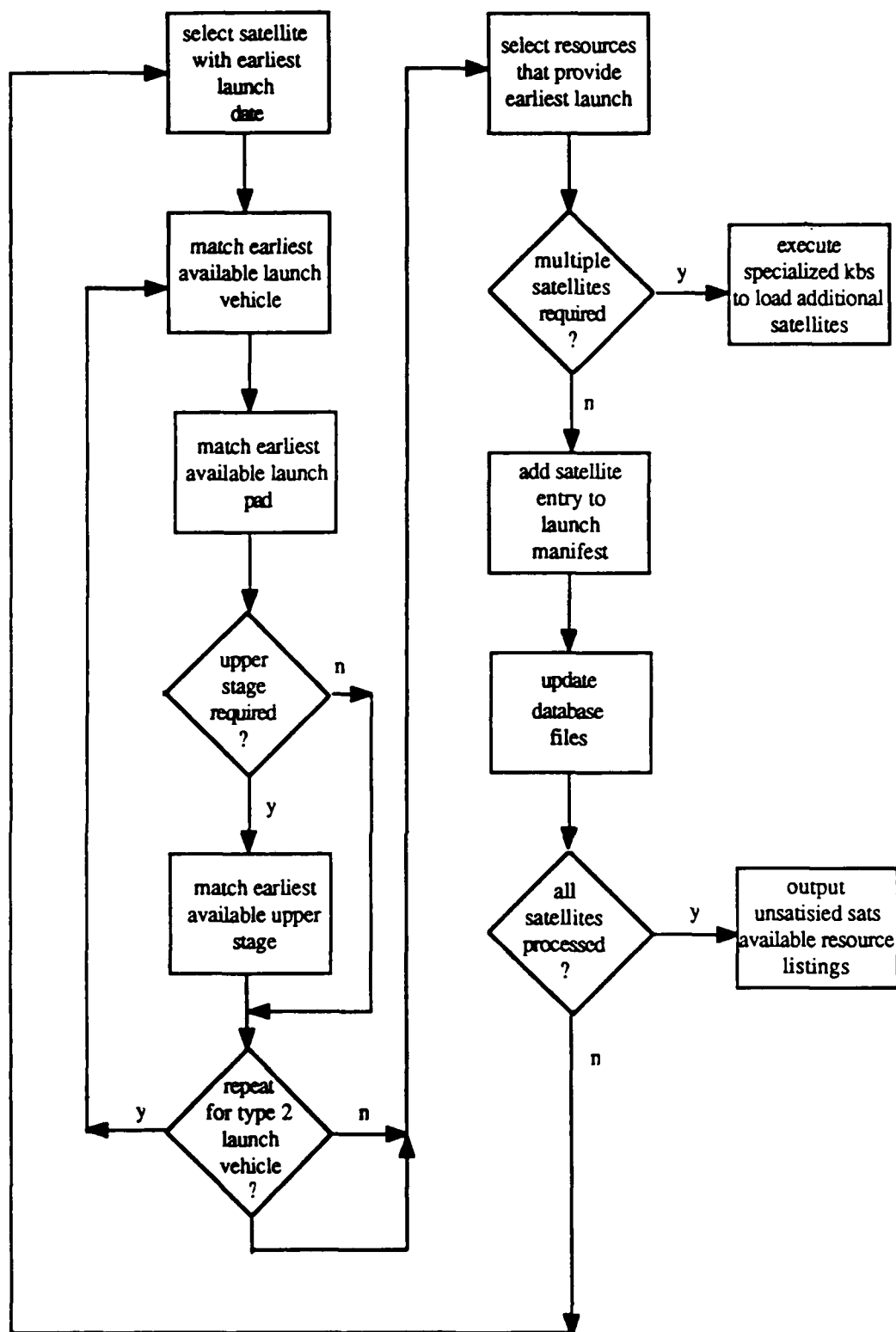


Figure 7. Manifesting Process

If the satellite requires an upper stage, then the Upper Stage fields of the satellite, launch vehicle, and launch pad records are matched against the Stage Type field of each upper stage record until the earliest available upper stage is found. This insures the upper stage can boost the satellite, the launch vehicle can accommodate the upper stage, and the launch pad can process the upper stage.

To determine the earliest day the satellite could be launched, the following calculation occurs. The pad Next Available day is the earliest day site processing can begin. If the launch vehicle is available before this day, then the launch vehicle Site Processing Time is added to the pad Next Available day to determine when upper stage processing can begin. If the launch vehicle is available after this earliest day, then the earliest day is advanced to the day the launch vehicle is available and launch vehicle Site Processing Time is added to determine when upper stage processing can begin. The same calculation occurs for the upper stage and the satellite until the earliest day all resources are available and processed for launch is determined.

Once a type 1 launch vehicle, launch pad, and upper stage are matched to the satellite, there is a potential entry for the manifest. However, it may not be the earliest available launch that meets the satellite requirement. To insure the satellite requirement is met by the earliest available resources, a type 2 launch vehicle, if any, must be matched against the satellite requirement. Launch pad and upper stage

are again matched, and then LRS-II selects the launch vehicle, launch pad, and upper stage combination that meets the satellite requirement earliest.

After the satellite requirement is met, LRS-II checks whether the satellite constellation and selected launch vehicle allow multiple satellites. If multiple satellites are allowed, LRS-II passes control to specialized knowledge bases to match additional satellites and upper stages. These specialized knowledge bases search the satellite database for the next unlaunched satellite (must be same constellation) and load the second satellite on the selected launch vehicle. If a second satellite or required upper stage are not available, the first satellite is launched by itself.

The combined site processing time to process the first satellite and second satellite (if loaded) determine the earliest day the satellite requirement can be met. Per USSPACECOM direction, if the Earliest Launch Date is earlier than the satellite Desired Launch Date, then the satellite is launched on the Desired Launch Date. If the Earliest Launch Date is later than the Desired Launch Date, then the satellite is launched on the Earliest Launch Date (Thompson, 1987:3).

After the satellite is added to the launch manifest, the satellite and launch resource records are updated. The Earliest Launch Date, scheduled Launch Date, IOC, resources used, and resource delay fields of the manifested satellite are updated. Each resource record is updated to show how the resource was used and when it is available again, if reusable.

The above steps are repeated until all satellite requirements are processed. If, during processing, launch resources for a type 1 launch vehicle are unavailable to meet the satellite requirement, LRS-II enters the reason in the Reason Satellite Not Launched 1 field of the satellite record and attempts to match a type 2 launch vehicle. If a type 2 launch vehicle is matched, LRS-II continues matching other resources. Again, if a resource is unavailable, the reason is listed in the Reason Satellite Not Launched 2 field of the satellite record and LRS-II attempts to process the next satellite requirement. LRS-II attempts to match launch resources to satellite requirements until all satellites are processed or a satellite Desired Launch Date exceeds the Ending Day of the schedule.

Output

The three outputs generated by LRS-II are the launch manifest, the list of unsatisfied satellite requirements, and the list of available launch resources at the end of the schedule.

An example of a launch manifest is shown in Figure 8. The top of the manifest shows who created it, when they created it, and any additional comments. The program header is followed by the starting and ending day of the schedule, and any special information about how to schedule particular satellite constellations. The rest of the manifest lists individual

Product created by Capt Ed Crawford on 9 Nov 87

Sample Manifest

L R S - I I
Launch Resource Scheduling
by
Capt Ed Crawford

Scheduling Period
Schedule GPS Multiply
Schedule TYPE1 Multiply

1 TO 999999

SATELLITE SCHEDULED	DESIRED DATE	EARLY DATE	LAUNCH DATE	IOC DATE	LAUNCH VEHICLE	LAUNCH PAD	UPPER STAGE	REASON MISSED
GPS-1	1	91	91	121	DELTA-1	SLC-17A	SGS-1	
GPS-2	2	101	101	131	DELTA-2	SLC-17B	SGS-2	
GPS-4	4	132	132	162	OV-1	SLC-39A	SGS-4	
GPS-3	3	132	132	162	OV-1	SLC-39A	SGS-3	
SOOS-1	5	124	124	154	SCOUT-1	SLC-5		
NOVA-1	6	228	228	258	SCOUT-2	SLC-5		
DMSP-1	7	0	0	0				NO VEHICLE

Figure 8. Launch Manifest

entries for each satellite scheduled. Each entry lists the satellite scheduled, satellite scheduling dates, launch resources used, and the reason the satellite missed launch, if required.

The list of unsatisfied satellite requirements is the second LRS-II output. Each entry in this list includes the satellite missing launch, the satellite desired launch date, and the reason the satellite missed launch.

The final output of LRS-II is a list of available launch resources at the end of the schedule. This output is made up of separate lists of available launch vehicles, launch pads, and upper stages and when they are available.

User Interface

The user interface is shown in Figure 9. LRS-II provides a menu-driven interface that allows the user to view database records on the screen or printer, reset database files, or start schedule generation. The main menu accesses sub-menus to allow the user to reset or view each individual database, all resource databases, or reset all database entries. Resetting the satellite database lists each satellite as available for launch by erasing previous data entries. Resetting the launch resource databases lists each resource available on its First Available date. This feature of selectively resetting databases allows the user to iteratively build a launch manifest.

LRS II : Launch Resource Scheduling V2.0

Choose the action you wish to perform:

- Reset DB Files
- View DB Files
- Change DB Files
- Generate Schedule
- Quit

Which files do you want to be Reset ?

- All DB Files
- All Resource DB Files
- Satellite DB Files
- Launch Vehicle DB Files
- Launch Pad DB Files
- Upper Stage DB Files
- Return to Main Menu

Figure 9. User Interface

Once the user selects Generate Schedule, LRS-II prompts the user for the manifesting period and other header information. LRS-II also prompts the user for single or multiple manifesting of satellites for each constellation as necessary. This allows the user to control how each satellite constellation is manifested.

LRS-II also allows the user to select two modes of scheduling. The first mode allows the user to specify the header information, the manifesting period, and the manifesting option for each satellite constellation. LRS-II then processes each satellite requirement without further user interaction. Each manifest entry is sent directly to a printer and a user specified file.

The second mode of scheduling allows the user to interact directly with the scheduling process. As each manifesting entry is generated, the output is displayed on the computer screen. The user can either accept the proposed matching by depressing the carriage return, or use Level 5 report functions to understand or change the line of reasoning used. This mode is ideal for debugging LRS-II modifications or to provide precise control of resource matching.

Development Iterations

Development of LRS-II went through several iterations. The initial system built met the prototype system requirements as presented in Chapter Five. The prototype system allowed the manifesting of any single satellite to a single launch vehicle.

The next development iteration added a specialized knowledge base to allow manifesting of multiple satellites for the GPS satellite constellation. This knowledge base schedules a single satellite if the matched launch vehicle is a Delta and schedules multiple satellites if the matched launch vehicle is a Shuttle.

The third iteration of LRS-II changed the manifest rules to insure the satellite always received the earliest available launch. This iteration was necessary due to a problem uncovered during testing. This problem is discussed in detail in the next chapter.

The final development iteration added the constraint of site processing time to the manifesting process. This allows LRS-II to determine the earliest launch date for each satellite. USSPACECOM can use this information to adjust launch dates if necessary. If a satellite scheduled for launch on its desired launch day impacts the launch of another satellite due to resource conflicts, then the satellite causing the impact can be launched between its earliest day and the desired day. This iteration also added detailed database updating and the remaining rules necessary to manifest all 13 satellite constellations correctly. During each iteration of development, testing verified LRS-II functions executed properly. A discussion of the test and evaluation of LRS-II are presented in the next chapter.

VIII. Test and Evaluation

Introduction

This chapter presents the test and evaluation of LRS-II. To verify and validate its correct operation, three levels of testing were used. The first level of testing verified the correct operation of each individual module. The second level of testing validated the integrated LRS-II system manifested each satellite constellation correctly. The third level of testing was an actual field test of the prototype LRS-II system. The results of each level of testing are presented next.

Module Level Testing

Module level testing was used throughout the development process to incrementally build the LRS-II system. The individual modules tested were the user interface, the manifesting knowledge base, and specialized knowledge bases to manifest multiple satellites.

To accomplish module testing, a series of test cases was used to iteratively refine the design of LRS-II. Each test case was designed to exercise a specific function of each module. The modules tested and functions exercised are shown in Table VI.

Table VI. TEST MATRIX

FUNCTION	1	2	3	4	5	6	7	8	9	10
User Interface										
Reset DB Files	X	X	X	X	X	X	X	X	X	X
View DB Files	X	X	X	X	X	X	X	X	X	X
Change DB Files	X									
Generate Schedule	X	X	X	X	X	X	X	X	X	X
Quit	X									
Schedule Choice	X	X	X	X	X	X	X	X	X	X
Manifesting KB										
Enter Schedule Info	X	X	X	X	X	X	X	X	X	X
Scheduling Mode	X	X	X	X	X	X	X	X	X	X
Set Run Step	X	X	X	X	X	X	X	X	X	X
Out Of Schedule Time										
Satellite NLT Exceeded	X	X		X						
Vehicle1 Not Available	X	X			X	X	X			
No Pad For Vehicle1	X	X		X						
No Stage For Vehicle1		X								
Vehicle2 Not Available	X	X	X	X	X	X	X		X	
No Pad For Vehicle2	X	X								
No Stage For Vehicle2	X	X								
All Satellites Processed	X	X		X	X	X	X	X	X	X
Find Available Satellite	X	X	X	X	X	X	X	X	X	X
Advance Schedule Day	X	X	X	X	X	X	X	X	X	X
Find Available Vehicle1	X	X	X	X	X	X	X	X	X	X
Find Available Vehicle2	X	X	X	X	X	X	X	X	X	X
Find Available Pad	X	X	X	X	X	X	X	X	X	X
Find Available Stage	X	X	X	X	X	X	X	X	X	X
Mark Projected Launch	X	X	X	X	X	X	X	X	X	X
Earliest Launch Vehicle1	X	X	X	X	X	X	X	X	X	X
Earliest Launch Vehicle2	X	X	X	X	X	X	X	X	X	X
Update Site Processing	X	X	X	X	X	X	X	X	X	X
Mark Missed Launch	X	X	X	X	X	X	X	X	X	X
Chain For GPS									X	X
Chain For Type1									X	X
Launch Satellite	X	X	X	X	X	X	X	X	X	X
Update Database Records	X	X	X	X	X	X	X	X	X	X
List Unlaunched Satellites	X	X		X	X	X	X			
List Available Resources	X	X	X	X	X	X	X	X	X	X
Specialized KBs										
Find Second Satellite									X	X
Find Second Stage									X	X
Launch Satellites									X	X

An individual test consisted of entering the satellite requirement and launch resource information into each database. Next, LRS-II operation was started and specific module functions were executed. The Level 5 reports system was used to verify that each rule of a function executed correctly. As errors were discovered, the Level 5 editor allowed easy modification of the rule and the test case was reaccomplished.

Module level testing allowed each function of a knowledge base to be developed and executed. As each function was successfully tested, the next function was added and this process was iteratively repeated until the complete LRS-II system was developed and ready for integrated testing.

System Level Testing

System level testing validated that the integrated LRS-II system manifested each satellite constellation correctly. Correctly means the proper launch vehicles, launch pads, and upper stages were matched against satellite requirements as specified by USSPACECOM.

The correct matching of launch resources for each satellite constellation is shown in Table VII. When a launch vehicle type is matched to the satellite, the remaining resource types are restricted by that choice. This is why it is important to schedule both launch vehicle types and then to manifest the launch resource configuration that meets the satellite requirement earliest.

Table VII. Launch Resource Configurations

SATELLITE	LAUNCH VEHICLE	LAUNCH PAD	UPPER STAGE
GPS	DELTA	SLC-17A	SGS
GPS	DELTA	SLC-17B	SGS
GPS	SHUTTLE	SLC-39A	SGS
GPS	SHUTTLE	SLC-39B	SGS
SOOS	SCOUT	SLC-5	
NOVA	SCOUT	SLC-5	
DMSP	ATLAS-E	SLC-3W	
DMSP	TITAN-2	SLC-4W	
NOAA	ATLAS-E	SLC-3W	
NOAA	TITAN-2	SLC-4W	
TDRS	SHUTTLE	SLC-39A	IUS
TDRS	SHUTTLE	SLC-39B	IUS
FLTSAT	ATLAS-G	SLC-36B	CENTAUR
TYPE1	TITAN-4	SLC-41	IUS
TYPE1	ATLAS-G	SLC-36B	CENTAUR
TYPE2	TITAN-4	SLC-41	IUS
TYPE3	TITAN-4	SLC-41	CENTAUR
TYPE4	TITAN-4	SLC-41	CENTAUR
TYPE4	SHUTTLE	SLC-39A	IUS
TYPE4	SHUTTLE	SLC-39B	IUS
TYPE5	TITAN-4	SLC-4E	
TYPE6	DELTA	SLC-17A	
TYPE6	DELTA	SLC-17B	

To accomplish system level testing, a 23 satellite test case was used to match each satellite constellation to every possible configuration of launch resources for that constellation. The results of this test validated LRS-II as ready for field testing at USSPACECOM.

Field Testing

Field testing demonstrated the actual operation of the LRS-II prototype to its intended user, USSPACECOM. The primary purpose of field testing was to allow the user to see the operation of the prototype system and to propose extensions that would make LRS-II more useful in its operational environment. These extensions are discussed in the Development Iterations section of Chapter VII. A secondary purpose of field testing was to use actual launch system data to test LRS-II in its operational environment.

Evaluation

The three levels of testing allowed the incremental development and validation of LRS-II. Module level testing allowed the iterative refinement of the initial system design. This level of testing uncovered two major flaws in the manifesting process.

In the initial design, the Launch Vehicle 1 and Launch Vehicle 2 fields of the satellite record were matched against the Vehicle Type field of each launch vehicle record until the earliest available launch vehicle was found. Next, the

earliest available launch pad and upper stage were matched to the selected launch vehicle and the configuration was manifested.

The problem was that the earliest available launch date may not have been selected. A launch pad or upper stage for the selected launch vehicle may not be available for two years. Whereas, the vehicle not selected may be available in two weeks and a compatible launch pad and upper stage are already available.

To correct this problem, both launch vehicle types are matched to available launch pads and upper stages and then the earliest available resource configuration is manifested by LRS-II.

Also in the initial design, the upper stage fields of the satellite record were matched against available upper stages. The problem was that although the upper stage matched to the satellite, it was not always compatible with the selected launch vehicle.

To correct this problem, upper stage fields in the satellite, launch vehicle, and launch pad records are matched against available upper stages until the earliest available upper stage is found. This insures the upper stage can boost the satellite, the launch vehicle can accommodate the upper stage, and the launch pad is modified to process the satellite. A summary of the research and recommendations are presented in the next chapter.

IX. Conclusions and Recommendations

Summary of Research

This research effort culminated with the extension of the LRS prototype into an operational system ready for field testing called LRS-II. USSPACECOM operators will use LRS-II as a decision aid to determine if there is sufficient launch capability to meet future satellite requirements and to quickly assess the impact of contingencies such as launch or on-orbit failures.

LRS-II uses multiple knowledge bases to match satellite launch requirements to available launch vehicles, launch pads, and upper stages. Specialized knowledge about satellite requirements and launch resources is stored in dBase III files. Knowledge base rules match specific fields of the satellite record against fields in the launch resource records to schedule the launch resources that meet the satellite requirement earliest. If two different types of launch vehicles are capable of launching a satellite, LRS-II builds resource paths using each launch vehicle and then selects the path that meets the requirement earliest. If the satellite constellation allows multiple satellites on a single launch vehicle, specialized knowledge bases are executed to manifest the additional satellites.

During the scheduling process, several constraints are considered which may impact the satellite launch date. Satellite and launch resource availability, site processing time, shuttle mission duration, and satellite on-orbit checkout time are applied to insure the selected launch date is accurate.

LRS-II provides detailed updating of database records when each launch is scheduled. Satellite updating includes recording the earliest day the satellite could be launched; recording the scheduled launch day and resources utilized; recording if resource availability delayed the launch past its desired date and which resources caused the delay; and recording the reason the satellite was not launched. In addition, each resource record is updated to show which satellite utilized it and, if reusable, when it is available again.

The outputs produced by LRS-II are the launch manifest, the list of unsatisfied satellite requirements, and the list of available launch resources. The launch manifest is a complete listing of the satellites scheduled for launch and the resources scheduled. The listing shows the satellite scheduled, the desired launch date, the earliest launch date, the actual launch date, the satellite IOC, and the launch vehicle, launch pad, and upper stage scheduled for launch. The listing also shows why the satellite missed launch if it could not be scheduled.

The menu-driven user interface allows the user to view database files on the screen or printer, reset all or selected database files, and start schedule generation. LRS-II also allows the user to select two modes of scheduling. Continuous scheduling allows the user to specify the manifesting period and scheduling choice (single or multiple) for each constellation. LRS-II then processes each satellite requirement without further user interaction. Each manifest entry is sent directly to a printer and a user specified file. Interactive scheduling allows the user to review and accept each manifest entry or to modify the proposed entry using the L5 report systems. This mode of scheduling is excellent for debugging LRS-II modifications or providing precise control of resource matching.

Recommendations

Before LRS-II is used operationally, it should be extensively field tested by USSPACECOM to insure the manifesting rules are sufficiently robust. USSPACECOM is gathering all pertinent launch systems data to prepare for field testing after LRS-II is delivered.

There are several extensions which could be implemented in future thesis research. These are applying Operational Research (OR) techniques to obtain an optimal schedule, using constraint directed search, scheduling satellites by priority, using month/day/year dates, and collecting statistics and presenting graphical displays of resource utilization.

A natural extension of LRS-II is to take the launch manifest generated and apply OR techniques to obtain an optimal schedule. LRS-II manifests each satellite on its desired launch date or on its earliest launch date if the earliest date is later than the desired date. The problem is that an earlier manifested launch may impact a later launch. This can be corrected by adjusting the launch date of the earlier manifested launch back towards its earliest launch date. Currently, USSPACECOM plans to do this manually by examining the LRS-II output and adjusting launch dates as necessary.

Another approach to the resource scheduling problem is to use constraint directed search to build the launch manifest. As discussed in the Literature Review, Fox has used this approach to solve the job shop scheduling problem. If each satellite requirement represents a job, then the availability of launch resources can be viewed as constraints on the schedule generation process. This method of scheduling may provide an optimal schedule.

LRS-II schedules satellites by earliest desired launch date. This speeds up execution because no record keeping of shuttle or pad utilization is required. However, it is often desirable to schedule satellites by priority to insure critical satellites are provided the earliest available resources. To do this using LRS-II, satellites must be ordered by earliest launch date and then the generated launch manifest must be

examined and reordered if necessary. A system which scheduled by priority would insure critical satellites are manifested first and would provide a detailed record of the utilization of reusable resources.

Using a month/day/year representation for dates would provide a more readable date than the Julian date representation. dBase III allows a date format; however, when the date is supplied to L5 it is represented as a string. dBase III has built-in functions which convert the string to an integer, but the number generated is not usable. To explain, if the date is entered as 10/23/87 the string is 19871023 and the dBase functions convert the string to the integer 19871023. This value may be modified during LRS-II processing, but only the last two digits represent the day. LRS-II may update the sixth digit if more than 99 days are added to a date, erroneously incrementing the month. Rules could be written to check for this condition before incrementing, and subroutines could be used to increment the integer date correctly. dBase functions could then convert the integer back to the month/day/year format.

A useful extension to LRS-II is to create graphical displays of resource utilization. All LRS-II outputs are listings which may be displayed on a printer or computer screen. It is difficult to determine resource utilization from these listings. Graphical displays would allow USSPACECOM operators to easily assess how resources are utilized.

Appendix A: LRS-II User Manual

Introduction

LRS-II version 1.0 is described in this manual. LRS-II was developed using Level 5 (L5) version 1.0 and dBase III. If a later version of L5 or dBase is used, refer to the current reference manual for any compatibility problems.

Required Equipment and Installation

LRS-II was developed on a Zenith Z-248 microcomputer with 640K RAM, a 2 megabyte RAM expansion board, one 720 kilobyte floppy disk drive, and a 20 megabyte hard disk drive. For hard disk installation follow the steps below :

1. Insert the L5 disk A: into the floppy drive.
2. Type *INSTALL* on the keyboard.
3. Follow the installation instructions as they appear on the screen.

It is recommended that LRS-II be loaded into a RAM disk if expanded RAM is available. A RAM disk is recommended due to the frequent disk accesses required during program operation.

To load LRS-II into a RAM disk follow the steps below :

1. Create an L5 and PRL directory in the RAM disk.
2. Copy all L5 files from the hard disk into the L5 directory.
3. Copy all LRS-II files into the PRL directory.
4. Follow the L5 Reference Manual to change the file path to search the RAM disk directories.
5. Follow the LRS-II Program Operation instructions to run LRS-II.
6. Remember to change paths back to the hard disk before exiting L5.

Required Files

In addition to the required L5 files loaded into the L5 directory, the following LRS-II files must be available in the PRL directory for LRS-II to operate properly :

EDLRS	.KNB	User Interface knowledge base
GENERATE	.KNB	Manifesting knowledge base
LOADGPS	.KNB	Specialized knowledge base for GPS
LOADTP1	.KNB	Specialized knowledge base for TYPE1
SATELITE	.DBF	Satellite requirement database file
VEHICLE	.DBF	Vehicle resource database file
PAD	.DBF	Launch Pad resource data base file
STAGE	.DBF	Upper Stage resource database file
SATELITE	.DBF	Satellite requirements database file

The source code files are listed below. These files are not required to run LRS-II but are required should program modifications become necessary.

EDLRS	.PRL	User Interface knowledge base
GENERATE	.PRL	Manifesting knowledge base
LOADGPS	.PRL	Specialized knowledge base for GPS
LOADTP1	.PRL	Specialized knowledge base for TYPE1

Program Operation

See the L5 Reference Manual for instructions on running L5. From the L5 main menu select Run Knowledge Base. Select EDLRS using the space bar or arrow keys to position the pointer in front of EDLRS and press the RETURN/ENTER key.

The LRS-II introduction screen will be displayed as soon as LRS-II has completed loading. The top line on the introduction screen will display the LRS-II program name and the current version number. You may view a brief description of LRS-II by pressing function key 1 PAGE. Pressing function key 3 STRT will start LRS program execution.

User Interface

LRS-II provides a menu-driven interface that allows the user to reset database files, view database records on the screen or printer, or start schedule generation. The LRS-II Main Menu is displayed after pressing the STRT function key, see Figure 1. From this menu all LRS-II functions may be selected. Each function is described below.

Reset DB Files. The Reset DB Files selection presents a sub-menu of the information which may be reset, see Figure 1. The sub-menu allows resetting All DB Files, All Resource DB Files, Satellite DB Files, Launch Vehicle DB files, Launch Pad DB Files, and Upper Stage DB Files. Resetting the satellite database lists each satellite as available for launch by erasing previous data entries. Resetting the launch resource databases lists each resource available on its First Available

date. This feature of selectively resetting databases allows the user to iteratively build a launch manifest.

View DB Files. The View DB Files selection presents a sub-menu of the information which may be viewed. The sub-menu allows viewing All DB Files, Satellite DB Files, Launch Vehicle DB Files, Launch Pad DB Files, and Upper Stage DB Files.

Choosing All DB Files from a sub-menu will display or reset the database information currently being used by LRS-II. This includes the information in the SATELITE.DBF, VEHICLE.DBF, PAD.DBF, and STAGE.DBF files. This information may be viewed on the computer screen, or sent to the printer. Choosing selected DB files from a sub-menu will display or reset the selected information in an individual database file. If no information is desired to be viewed or reset, then choosing Return to Main Menu will return the user to the LRS-II Main Menu.

Change DB Files. The Change DB Files selection displays on the computer screen an explanation of how database files may be modified. To modify a database file, dBase must be executed from the L5 main menu or off-line. The modified files must be copied back into the PRL directory before LRS-II execution.

Generate Schedule. Generate Schedule starts matching the satellite requirements against available launch resources. When Generate Schedule is selected a brief explanation of the matching process of LRS-II is displayed. Pressing function key 2 CONT will start program execution.

The first action taken is for the user to enter their *name*, the *date*, and a line of *comment*. This information is printed in the header of the generated launch manifest and sent to the SCHEDULE.DAT file to identify the saved output file. LRS-II also prompts the user for the *scheduling choice* of single or multiple manifesting for each constellation as necessary. This allows the user to specify how LRS-II should manifest each satellite constellation. Currently, LRS-II only queries for *scheduling choice* for GPS and TYPE1 satellite constellations. For GPS, if the selected launch vehicle is a shuttle and *scheduling choice* is multiply, then two GPS satellites are manifested, if available. For TYPE1, if the selected launch vehicle is a Titan 4 and *scheduling choice* is multiply, then two TYPE1 satellites are manifested, if available. Otherwise, only a single satellite is manifested to a selected launch vehicle.

Choosing the *scheduling mode* is the next action. The choices are interactive scheduling by displaying the output on the computer screen or automatic scheduling (non-interactive) by displaying the output on the printer. Displaying output on the screen requires pressing function key 2 CONT after each manifest entry is presented. This mode of scheduling allows the user to interact directly with the scheduling process. As each manifesting entry is generated the output is displayed on the computer screen. The user can either accept the proposed matching by depressing the RETURN/ENTER key, or use L5 report functions to understand or change the line of reasoning used.

This mode is ideal for debugging LRS-II modifications or to provide precise control of resource matching.

Displaying output on the printer will send each manifest entry directly to the printer requiring no user interaction during the matching process. The user specifies the *header* information, the *scheduling choice*, the *scheduling mode*, and the *manifesting period*. LRS-II then processes each satellite requirement without further interaction. Regardless of the scheduling mode, the schedule information is also sent to the SCHEDULE.DAT file. This file may be viewed, printed, or saved for later use, after exiting LRS-II. IMPORTANT !! The information stored in SCHEDULE.DAT is overwritten at the beginning of each LRS-II session.

The *manifesting period* information is entered next. This is the start and ending schedule day to be used. The start day may be any number between 1 and 999999. The ending day must be greater than or equal to the start day, or 0 to process all satellite requirements. LRS-II uses the Julian date format for date representation. These dates must be mapped to a month/day/year format for correct interpretation.

Once the schedule information is entered LRS-II will attempt to match the earliest available launch vehicle, launch pad, and upper stage to the satellite requirement. As each requirement is processed the manifest entry is sent to the selected output device and the next requirement is processed. LRS-II attempts to match resources to requirements until all satellites are processed or the ending schedule day is reached.

The requirement to resource matching process is explained in detail in the Manifesting Process section of the LRS-II User's Manual.

Quit. The Quit option ends the LRS-II session. An END OF SESSION message screen is displayed by L5. Press function key 8 MENU to return to the L5 Main Menu. Pressing Function key 10 EXIT will exit L5 and return the user to the DOS prompt.

Database Input Formats

All satellite requirement and launch resource knowledge is stored in dBase III database files. This section describes the format for each type of record and the type of knowledge required.

Satellite Record Format

The format of a satellite requirement record is shown in Table 1. A detailed explanation of each field is provided below.

Satellite Name. Satellite Name specifies the chosen designation for a particular satellite. This knowledge is used to identify the satellite manifested.

Constellation. Constellation is the name of the constellation the satellite is a member of. This knowledge is used to determine if the satellite should be manifested singly or multiply.

Launch Vehicle 1. Launch Vehicle 1 is the first type of launch vehicle which can launch the satellite. The Launch Vehicle 1 field must exactly match the Vehicle Type field of a launch vehicle record.

Launch Vehicle 2. Launch Vehicle 2 is the second type of launch vehicle which can launch the satellite. If a second type of launch vehicle is not available, then this field must be set to NONE.

Upper Stage 1. Upper Stage 1 is the first type of upper stage which can boost the satellite to its final operational orbit. The Upper Stage 1 field must exactly match the Stage Type field of an upper stage record, if the satellite requires an upper stage. If no upper stage is required, then this field must be set to NONE.

Upper Stage 2. Upper Stage 2 is the second type of upper stage which can boost the satellite. If a second type of upper stage is not available, then this field must be set to NONE.

Coast. Coast identifies whether the selected launch pad must be on the EAST coast or WEST coast. Coast must exactly match the Coast field of the selected launch pad record.

Available. Available is the Julian date the satellite is ready for site processing. This date must be any integer between 1 and 999999.

Site Processing Time. Site Processing Time is the number of days it takes to process the satellite once it reaches the launch pad. This number must be an integer between 1 and 999.

Desired Launch Date. Desired Launch Date is the earliest Julian date the user desires to launch the satellite. The user must order the satellite database by earliest Desired Launch Date to insure the satellite requiring launch earliest is matched against the entire pool of available resources. This establishes a priority based on Desired Launch Date. This date must be an integer between 1 and 999999.

NLT Launch Date. NLT Launch Date is the latest Julian date the user desires to launch the satellite by. If the satellite is not launched with all required launch resources by this day, it is considered as missing its launch and the appropriate Reason Satellite Not Launched field of the satellite record is marked NLT EXCEEDED. This date must be an integer between 1 and 999999. The number of days between the Desired Launch Date and the NLT Launch Date determines the launch window for the satellite.

On-Orbit Checkout Time. On-Orbit Checkout Time is the number of days it takes to checkout the launched satellite before it is considered operational. This number must be an integer between 1 and 999.

Launched. Launched is the launch status of the satellite. If the value is "N" or "X" the satellite is not launched and is available for scheduling. If the value is "L"

the satellite was launched and LRS-II skips to the next satellite record. When the satellite database is reset, the Launched field is set to "N" to indicate all satellites are available for launch. During LRS-II processing the Launched field is set to "X" if a satellite cannot be launched due to resource unavailability or NLT exceeded, or Launched is set to "L" to indicate the satellite was manifested.

Earliest Launch Date. Earliest Launch Date is the earliest possible Julian date the satellite could be launched, as calculated by LRS-II. This date is calculated by using the Next Available field of the matched launch pad record to determine when site processing can begin. Site processing consists of processing the launch vehicle, an upper stage (if required), and the satellite. The Next Available and Site Processing Time fields of the matched records are used to determine the Earliest Launch Date. The earliest date all resources are available and processed is the Earliest Launch Date for the satellite.

Launch Date. Launch Date is the actual Julian date the satellite is scheduled for launch. This date is the Desired Launch Date if the Desired Launch Date is later than the Earliest Launch Date, or this date is the Earliest Launch Date if the satellite cannot be launched by the Desired Launch Date.

IOC. IOC is the Launch Date plus the On-Orbit Checkout Time.

Vehicle Used. Vehicle Used is the selected launch vehicle scheduled to launch the satellite.

Pad Used. Pad Used is the selected launch pad scheduled to launch the satellite.

Stage Used. Stage Used is the selected upper stage scheduled to launch the satellite.

Vehicle Delay. Vehicle Delay set to "Y" indicates the launch vehicle availability exceeded the Desired Launch Date.

Pad Delay. Pad Delay set to "Y" indicates the launch pad availability exceeded the Desired Launch Date.

Stage Delay. Stage Delay set to "Y" indicates the upper stage availability exceeded the Desired Launch Date.

Reason Satellite Not Launched 1. Reason Satellite Not Launched 1 is set to NO VEHICLE, NO PAD, NO STAGE, or NLT EXCEEDED if the satellite misses launch when LRS-II attempts to match a launch vehicle, launch pad, and upper stage using a Launch Vehicle 1 type launch vehicle.

Reason Satellite Not Launched 2. Reason Satellite Not Launched 2 is set to NO VEHICLE, NO PAD, NO STAGE, or NLT EXCEEDED if the satellite misses launch when LRS-II attempts to match a launch vehicle, launch pad, and upper stage using a Launch Vehicle 2 type launch vehicle.

Launch Vehicle Record Format

The format of a launch vehicle resource record is shown in Table II. A detailed explanation of each field is provided below.

Vehicle Name. Vehicle Name specifies the chosen designation for a particular launch vehicle. This knowledge is used to identify the launch vehicle utilized.

Vehicle Type. Vehicle Type is the type of launch vehicle. This field matches the launch vehicle to the satellite and the launch pad. The Vehicle Type must exactly match the Launch Vehicle fields of the satellite record and the Pad Type field of the pad record.

East Pad 1. East Pad 1 is the first type of launch pad on the east coast which can launch the launch vehicle. The East Pad 1 field must exactly match the Pad Name of the pad record, or set to NONE1 if an east coast pad is not available.

East Pad 2. East Pad 2 is the second type of launch pad on the east coast which can launch the launch vehicle. The East Pad 2 field must exactly match the Pad Name of the pad record, or set to NONE1 if a second east coast pad is not available.

West Pad 1. West Pad 1 is the first type of launch pad on the west coast which can launch the launch vehicle. The West Pad 1 field must exactly match the Pad Name of the pad record, or set to NONE1 if a west coast pad is not available.

West Pad 2. West Pad 2 is the second type of launch pad on the west coast which can launch the launch vehicle. The East Pad 1 field must exactly match the Pad Name of the pad record, or set to NONE1 if a second west coast pad is not available.

Upper Stage 1. Upper Stage 1 is the first type of upper stage which can fly on the launch vehicle. The Upper Stage 1 field must exactly match the Stage Type field of an upper stage record, if the launch vehicle requires an upper stage. If no upper stage is required, then this field must be set to NONE.

Upper Stage 2. Upper Stage 2 is the second type of upper stage which can fly on the launch vehicle. If a second type of upper stage is not required, then this field must be set to NONE1.

First Available. First Available is the Julian date the launch vehicle is first available. This date is used to reset the Next Available field when the launch vehicle database is reset.

Site Processing Time. Site Processing Time is the number of days it takes to process the launch vehicle once it reaches the launch pad. This number must be an integer between 1 and 999.

Mission Duration. Mission Duration is the number of days a shuttle is in orbit. For shuttles, this number must be an integer between 1 and 999. For ELVs this number must be 0.

AD-A194 625

LRS-II: A SPECIALIZED KNOWLEDGE SYSTEM FOR LAUNCH
RESOURCE SCHEDULING(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING

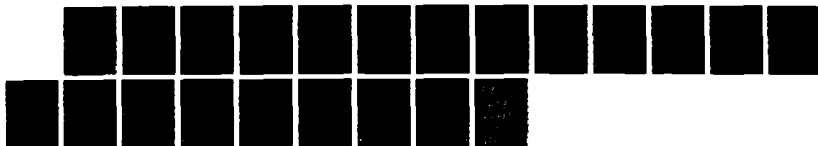
2/2

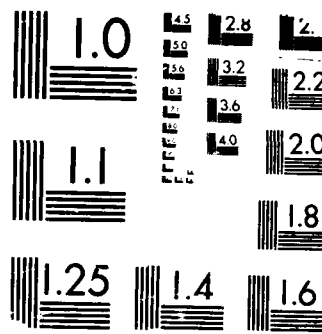
UNCLASSIFIED

J E CRAWFORD DEC 87 AFIT/GE/ENG/87D-11

F/G 5/1

NL





MICROCOPY RESOLUTION TEST CHART
 NATIONAL BUREAU OF STANDARDS-1963-A

Turn Time. Turn Time is the number of days it takes to refurbish a shuttle after launch. The Turn Time and Mission Duration are added to the Launch Date to determine when a shuttle will be Next Available for launch. For shuttles, this number must be an integer between 1 and 999. For ELVs this number must be 0.

Launch Date. Launch Date is the Julian date the launch vehicle was utilized.

First Satellite. First Satellite is the first satellite scheduled for launch on the selected launch vehicle.

Second Satellite. Second Satellite is the second satellite scheduled for launch on the selected launch vehicle, when multiple manifesting of satellites is allowed.

Next Available. Next Available is the Julian date a shuttle launch vehicle can be reused. For ELVs, this date is set to zero.

Launch Pad Record Format

The format of a launch pad resource record is shown in Table III. A detailed explanation of each field is provided below.

Pad Name. Pad Name specifies the chosen designation for a particular launch pad. This knowledge is used to identify the launch pad utilized.

Pad Type. Pad Type is the type of launch pad. This field matches the launch vehicle to the launch pad. The Pad Type must exactly match the Vehicle Type field of the launch vehicle record.

Coast. Coast identifies whether the launch pad is on the EAST coast or WEST coast. Coast must exactly match the Coast field of the satellite record.

Upper Stage 1. Upper Stage 1 is the first type of upper stage which the launch pad can process. The Upper Stage 1 field must exactly match the Stage Type field of an upper stage record, if the satellite requires an upper stage. If no upper stage is required, then this field must be set to NONE.

Upper Stage 2. Upper Stage 2 is the second type of upper stage which the launch pad can process. If a second type of upper stage is not available, then this field must be set to NONE2.

First Available. First Available is the Julian date the launch pad is first available. This date is used to reset the Next Available field when the launch pad database is reset.

Turn Time. Turn Time is the number of days it takes to refurbish the launch pad after launch. This number must be an integer between 1 and 999.

Next Available. Next Available is the Julian date the launch vehicle can be reused. This date is equal to the Launch Date plus the Turn Time.

Upper Stage Record Format

The format of an upper stage resource record is shown in Table IV. A detailed explanation of each field is provided below.

Stage Name. Stage Name specifies the chosen designation for a particular upper stage. This knowledge is used to identify the upper stage utilized.

Stage Type. Stage Type is the type of upper stage. This field matches the upper stage to the satellite, launch vehicle, and launch pad. The Stage Type must exactly match the Upper Stage fields of the satellite, launch vehicle, and launch pad records. This insures the upper stage can boost the satellite, the launch vehicle can carry the upper stage, and the launch pad can process the upper stage.

First Available. First Available is the Julian date the upper stage is first available. This date is used to reset the Next Available field when the upper stage database is reset.

Site Processing Time. Site Processing Time is the number of days it takes to process the upper stage once it reaches the launch pad. This number must be an integer between 1 and 999.

Launch Date. Launch Date is the Julian date the upper stage was utilized.

Satellite Boosted. Satellite Boosted is the satellite scheduled for launch using the selected upper stage.

TABLE I.

Database format information for the SATELITE database

Field	Name	Type	Width
1	NAME	C	30
2	CONSTEL	C	30
3	LCH_VEH_1	C	30
4	LCH_VEH_2	C	30
5	U_S_1	C	30
6	U_S_2	C	30
7	COAST	C	30
8	AVAIL	N	6
9	SITE_PROC	N	3
10	DESIRE_LCH	N	6
11	NLT_LCH	N	6
12	C_O_TIME	N	3
13	LAUNCHED	C	1
14	LCH_DATE	N	6
15	IOC	N	6
16	VEH_USED	C	30
17	PAD_USED	C	30
18	STG_USED	C	30
19	VEH_DELAY	C	1
20	PAD_DELAY	C	1
21	STG_DELAY	C	1
22	MISSED1	C	30
23	MISSED2	C	30

TABLE II.

Database format information for the VEHICLE database

Field	Name	Type	Width
1	NAME	C	30
2	TYPE	C	30
3	PAD_1E	C	30
4	PAD_2E	C	30
5	PAD_1W	C	30
6	PAD_2W	C	30
7	U_S_1	C	30
8	U_S_2	C	30
9	FIRST_AVAL	N	6
10	SITE_PROC	N	3
11	MSN_DUR	N	3
12	TURN_TIME	N	3
13	LCH_DATE	N	6
14	SAT_1	C	30
15	SAT_2	C	30
16	NEXT_AVAL	N	6

TABLE III.

Database format information for the PAD database

Field	Name	Type	Width
1	NAME	C	30
2	TYPE	C	30
3	COAST	C	30
4	U_S_1	C	30
5	U_S_2	C	30
6	FIRST_AVAL	N	6
7	TURN_TIME	N	3
8	NEXT_AVAL	N	6

TABLE IV.

Database format information for the STAGE database

Field	Name	Type	Width
1	NAME	C	30
2	TYPE	C	30
3	FIRST_AVAL	N	6
4	SITE_PROC	N	3
5	LCH_DATE	N	6
6	SATELLITE	C	30
7	NEXT_AVAL	N	6

Using Multiple Data Base Files

LRS-II accesses the same database files for each session. To use different files for a session, the files to be used must be renamed to the file names used by LRS-II. The satellite requirement file is named SATELITE.DBF. The launch vehicle resource file is named VEHICLE.DBF. The launch pad resource file is named PAD.DBF. The upper stage resource file is named STAGE.DBF.

The easiest way to create a new database file is to copy an existing file and edit the records to make the new file. In this way there is no danger of accidentally using the wrong file format. The file format is critical to proper LRS-II operation.

LRS II : Launch Resource Scheduling V2.0

Choose the action you wish to perform:

MMM Reset DB Files

View DB Files

Change DB Files

Generate Schedule

Quit

Which files do you want to be Reset ?

MMM All DB Files

All Resource DB Files

Satellite DB Files

Launch Vehicle DB Files

Launch Pad DB Files

Upper Stage DB Files

Return to Main Menu

Figure 1. User Interface

Manifesting Process

LRS-II uses four knowledge bases to allow complete manifesting of all 13 constellations. However, a single knowledge base does all scheduling of a single satellite to a launch vehicle. This manifesting process is shown in Figure 2.

Processing begins with LRS-II selecting the satellite requirement with the earliest Desired Launch Date from the satellite database (the database must be ordered). If the satellite is marked as Launched, LRS-II selects the next satellite until it finds an unsatisfied satellite requirement.

The next step is to match the earliest available launch vehicle to the satellite. This is done by matching the Launch Vehicle 1 field of the satellite record against the Vehicle Type field of each launch vehicle record until the earliest available type 1 launch vehicle is found. The Upper Stage fields of the satellite and launch vehicle are also matched to insure the selected vehicle can accommodate the required upper stage.

Next, the earliest available launch pad is matched to the selected type 1 launch vehicle. This is done by matching the Pad fields of the launch vehicle record against the Pad Type field of each launch pad record until the earliest available launch pad is found. The Coast field of the satellite record and the pad record are also matched to insure a pad on the correct coast is selected.

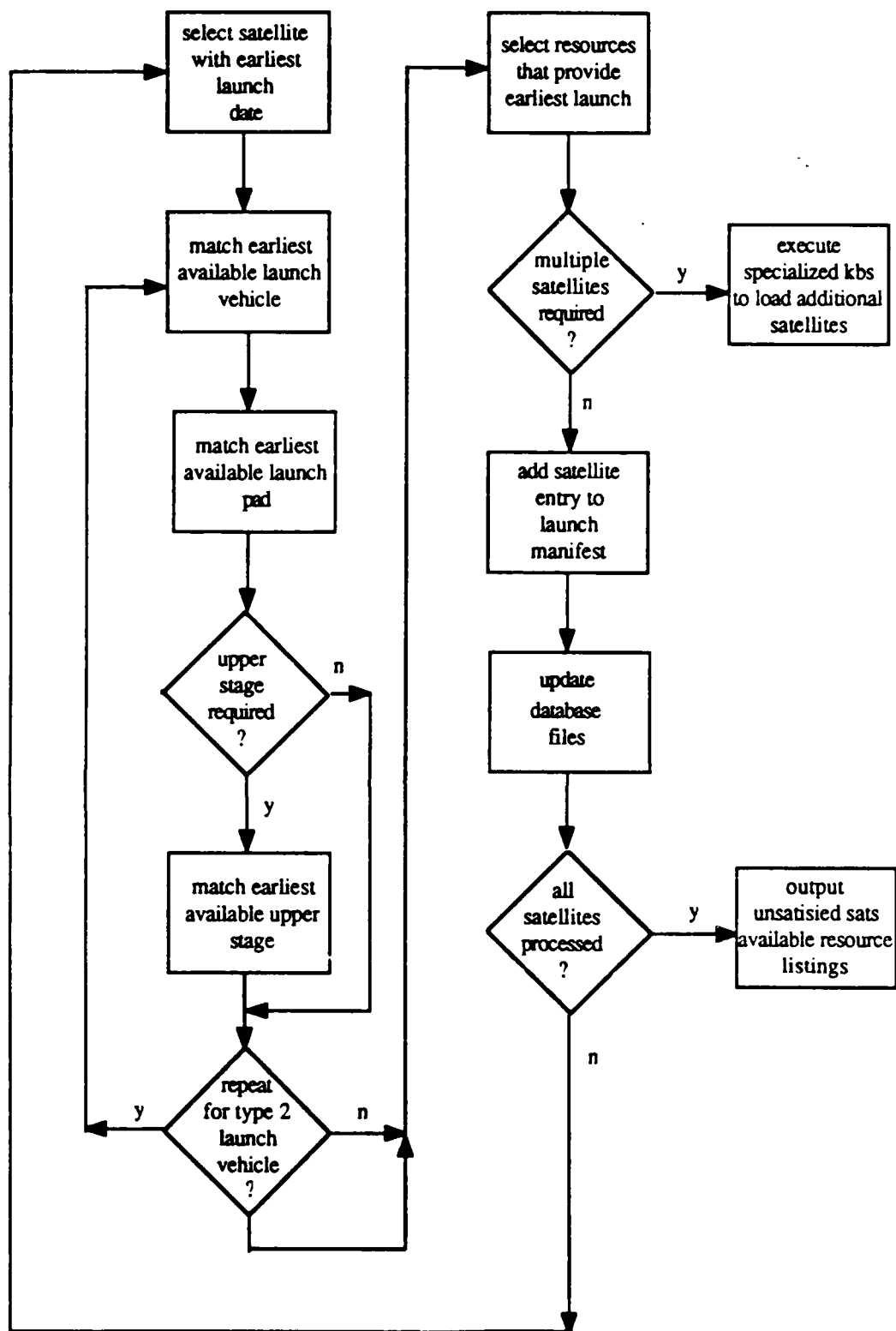


Figure 2. Manifesting Process

If the satellite requires an upper stage, then the Upper Stage fields of the satellite, launch vehicle, and launch pad records are matched against the Stage Type field of each upper stage record until the earliest available upper stage is found. This insures the upper stage can boost the satellite, the launch vehicle can accommodate the upper stage, and the launch pad can process the upper stage.

To determine the earliest day the satellite could be launched, the following calculation occurs. The pad Next Available day is the earliest day site processing can begin. If the launch vehicle is available before this day, then the launch vehicle Site Processing Time is added to the pad Next Available day to determine when upper stage processing can begin. If the launch vehicle is available after this earliest day, then the earliest day is advanced to the day the launch vehicle is available and launch vehicle Site Processing Time is added to determine when upper stage processing can begin. The same calculation occurs for the upper stage and the satellite until the earliest day all resources are available and processed for launch is determined.

Once a type 1 launch vehicle, launch pad, and upper stage are matched to the satellite, there is a potential entry for the manifest. However, it may not be the earliest available launch that meets the satellite requirement. To insure the satellite requirement is met by the earliest available resources, a type 2 launch vehicle, if any, must be matched against the satellite requirement. Launch pad and upper stage

are again matched, and then LRS-II selects the launch vehicle, launch pad, and upper stage combination that meets the satellite requirement earliest.

After the satellite requirement is met, LRS-II checks whether the satellite constellation and selected launch vehicle allow multiple satellites. If multiple satellites are allowed, LRS-II passes control to specialized knowledge bases to match additional satellites and upper stages. These specialized knowledge bases search the satellite database for the next unlaunched satellite (must be same constellation) and load the second satellite on the selected launch vehicle. If a second satellite or required upper stage are not available, the first satellite is launched by itself.

The combined site processing time to process the first satellite and second satellite (if loaded) determine the earliest day the satellite requirement can be met. Per USSPACECOM direction, if the Earliest Launch Date is earlier than the satellite Desired Launch Date, then the satellite is launched on the Desired Launch Date. If the Earliest Launch Date is later than the Desired Launch Date, then the satellite is launched on the Earliest Launch Date (Thompson, 1987:3).

After the satellite is added to the launch manifest, the satellite and launch resource records are updated. The Earliest Launch Date, scheduled Launch Date, IOC, resources used, and resource delay fields of the manifested satellite are updated. Each resource record is updated to show how the resource was used and when it is available again if reusable.

The above steps are repeated until all satellite requirements are processed. If, during processing, launch resources for a type 1 launch vehicle are unavailable to meet the satellite requirement, LRS-II enters the reason in the Reason Satellite Not Launched 1 field of the satellite record and attempts to match a type 2 launch vehicle. If a type 2 launch vehicle is matched, LRS-II continues matching other resources. Again, if a resource is unavailable, the reason is listed in the Reason Satellite Not Launched 2 field of the satellite record and LRS-II attempts to process the next satellite requirement. LRS-II attempts to match launch resources to satellite requirements until all satellites are processed or a satellite Desired Launch Date exceeds the Ending Day of the schedule.

Output

The three outputs generated by LRS-II are the launch manifest, the list of unsatisfied satellite requirements, and the list of available launch resources at the end of the schedule.

An example of a launch manifest is shown in Figure 3. The top of the manifest shows who created it, when they created it, and any additional comments. The program header is followed by the starting and ending day of the schedule, and any special information about how to schedule particular satellite constellations. The rest of the manifest lists individual

Product created by USSPACECOM/J30 on 23 SEPT 87

OPERATIONAL TEST

L R S - 1 1

Launch Resource Scheduling

by

Capt Ed Crawford
Major Fred Koch

Program Assumptions

This program uses all deterministic computations. This means if a vehicle launches it always successfully places its payload into orbit. If a satellite takes two weeks checkout time it will always take exactly two weeks to checkout.

All specific parameters for vehicles, stages, pads, and satellites are entered by the user. This information is used by the program to provide a possible matching of launch resources to launch requirements.

Satellites are scheduled using the desired launch date entered by the user. The satellite with the earliest desired launch date has the highest priority.

The schedule provided is not necessarily the optimum schedule. It is a feasible schedule. One which meets all the constraints established by the user entered information and the scheduling rules of the program.

The Starting Day of the Schedule is 1.

The Ending Day of the Schedule is 999999.

Schedule GPS Singly

SATELLITE SCHEDULED	DESIRED DATE	LAUNCH DATE	IOC DATE	LAUNCH VEHICLE	LAUNCH PAD	UPPER STAGE	REASON MISSED
SANMARCO-D	334	334	364	SCOUT-1	KENYA		
NOAA-H	414	414	444	A-63E	SLC-3W		
DMSP F-9	425	504	534	A-59E	SLC-3W		
TDRS-C	488	488	518	DISCOVERY	SLC-39A	IUS-1	
SOOS-3	516	516	546	SCOUT-2	SLC-5		
GPS-1	608	608	638	DELTA-1	SLC-17A	PAM-1	
COBE	700	700	730	DELTA-2	SLC-2		
TDRS-D	850	850	880	DISCOVERY	SLC-39A	IUS-2	
HUB TELE	942	942	972	ATLANTIS	SLC-39A		
ITV-2	1000	1000	1000	SCOUT-3	WALLOPS		

Figure 3. Launch Manifest

entries for each satellite scheduled. Each entry lists the satellite scheduled, satellite scheduling dates, launch resources used, and the reason the satellite missed launch, if required.

The list of unsatisfied satellite requirements is the second LRS-II output. Each entry in this list includes the satellite missing launch, the satellite desired launch date, and the reason the satellite missed launch.

The final output of LRS-II is a list of available launch resources at the end of the schedule. This output is made up of separate lists of available launch vehicles, launch pads, and upper stages and when they are available.

Bibliography

- Boller, Lt Col Rick. Chief, Control, Launch, and Support Systems Division. Personal Interview. USSPACECOM/J30, Peterson AFB CO, June 1987.
- Dutry, Maj Steve. Telephone Interview. Space Transportation System Directorate, Space Division, Los Angeles CA, April 1987.
- Foonberg, Allen M. SUMS - A Shuttle Utilization Management System. Report to Space Division, Los Angeles CA, January 1986.
- . Telephone Interview. Aerospace Corporation, Los Angeles CA, May 1987.
- Fox, Mark S. Constraint-Directed Search: A Case Study of Job-Shop Scheduling. PhD dissertation, Computer Science Department, Carnegie-Mellon University, Pittsburgh PA, December 1983.
- Kennedy, Capt Dale J. A Prototype Expert System Advisor For Satellite Support Scheduling. MS thesis, AFIT/GOR/OS/86D-5. School of Engineering, AFIT/ENS, Wright-Patterson AFB OH, December 1986.
- Koch, Maj Fred H. LRS: A Knowledge Based Approach to Launch Resource Scheduling. MS thesis, AFIT/GSO/ENS/86D-4. School of Engineering, AFIT/ENS, Wright-Patterson AFB OH, December 1986.
- Perales, Capt Dave. A Prototype Knowledge Based System For Satellite Mission Planning. MS thesis, AFIT/GE/ENG/86D. School of Engineering, AFIT/ENG, Wright-Patterson AFB OH, December 1986.
- Rowell, Lt Col. Professor, Department of Operational Sciences. Personal Interview. AFIT/ENS, Wright-Patterson AFB OH, May 1986.
- Thompson, Capt Norm. Correspondence. USSPACECOM/J30, Peterson AFB CO, April 1987.
- Waterman, Donald A. A Guide to Expert Systems. Addison-Wesley Publishing, 1986.
- Williams, Capt John E. Military Satellite Constellation Design. Report to Lt Col Greg Parnell. AFIT/ENS, Wright-Patterson AFB OH, 11 March 1987.
- Williams, Capt Tony. Stoplight. Report to USSPACECOM. AFSPACECOM/DOS, Peterson AFB CO, June 1987.

Additional Sources

dBase III User's Manual. Version 2.41. Ashton-Tate, Inc.,
Culver City CA, 1987.

Level 5 Reference Manual. Version 1.0. Level 5 Research, Inc.,
Indianalantic FL, 1987.

MS-DOS User's Guide. Version 2.11. Zenith Data Systems, 1985.

Personal Consultant Plus User's Guide. Version 2.0. Texas
Instruments, Inc., Austin TX, 1986.

PC Scheme User's Guide. Version 2.0. Texas Instruments, Inc.,
Austin TX, 1986.

VITA

Captain James E. Crawford Jr. was born on 30 December 1957 in New Orleans, Louisiana. He graduated from Chalmette High School in Chalmette, Louisiana in 1974 and attended The University of Texas at Austin from which he received the degree of Bachelor of Science in Electrical Engineering in December 1982. Upon graduation he received a commission in the USAF through the AECP program.

Captain Crawford entered the USAF in April 1976 and served seven years as an enlisted Airman before his commissioning. Capt Crawford's assignments include a four year tour as a Disbursement Accounting Specialist at RAF Mildenhall, United Kingdom, three years of study at the University of Texas, and a three year tour as an Atlas Launch Controller at Vandenberg AFB California. As an Atlas Launch Controller he was directly involved in the successful launch of 11 space missions. Captain Crawford entered the School of Engineering, Air Force Institute of Technology, in May 1986.

Permanent address: 4228 S. California Ave.
Chicago, IL 60632

A19-625

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE			Approved for public release; distribution unlimited		
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT / GE / ENG / 87D-11			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION School of Engineering		6b. OFFICE SYMBOL (If applicable) AFIT / ENG	7a. NAME OF MONITORING ORGANIZATION		
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433			7b. ADDRESS (City, State, and ZIP Code)		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code)			10. SOURCE OF FUNDING NUMBERS		
		PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) IRS-II: A SPECIALIZED KNOWLEDGE SYSTEM FOR LAUNCH RESOURCE SCHEDULING (U)					
12. PERSONAL AUTHOR(S) James E. Crawford, Jr., B.S., M.S., Capt, USAF					
13a. TYPE OF REPORT Thesis		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Year, Month, Day) 1987 December	
15. PAGE COUNT 105					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP			
12	07		Artificial Intelligence, Expert Systems,		
22	04		Space Operations, Scheduling Systems		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
Title: LRS-II: A SPECIALIZED KNOWLEDGE SYSTEM FOR LAUNCH RESOURCE SCHEDULING					
Thesis Chairman: Gregory S. Parnell, Lt Col, USAF Professor of Space Operations					
(See reverse for Abstract.)					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Gregory S. Parnell, Lt Col, USAF			22b. TELEPHONE (Include Area Code) 513-255-3362		22c. OFFICE SYMBOL AFIT / ENG

Approved for public release: 1AW AFR 100-17.
14 Mar 88
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AFIT)
Wright-Patterson AFB OH 45433

Abstract

This research used the Level 5 expert system software to develop a specialized knowledge system called the Launch Resource Scheduling system (LRS-II). LRS-II will be used as a decision aid by USSPACECOM to determine if there is sufficient launch capability to meet future satellite requirements and to quickly assess the impact of contingencies such as launch or on-orbit failures.

LRS-II uses multiple knowledge bases to match satellite launch requirements to available launch vehicles, launch pads, and upper stages. Specialized knowledge about satellite requirements and launch resources are stored in dBase III files. Level 5 knowledge base rules match specific fields of the satellite record against fields in the resource records to schedule the earliest launch resources that meet the satellite requirement. If two different types of launch vehicles are capable of launching a satellite, LRS-II selects the launch vehicle that meets the requirement earliest. If the satellite constellation allows multiple satellites on a single launch vehicle, specialized knowledge bases are executed to manifest the additional satellites. During manifesting, the constraints of satellite and resource availability, site processing time, shuttle mission duration, and satellite on-orbit checkout time are used to insure the selected launch date is accurate. (Thurs)

END

DATE

FILMED

8-88

DTIC